

A Heterogeneous Stochastic FEM Framework for Elliptic PDEs

Thomas Y. Hou*

Pengfei Liu†

Abstract

We introduce a new concept of sparsity for the stochastic elliptic operator $-\operatorname{div}(a(x, \omega) \nabla(\cdot))$, which reflects the compactness of its inverse operator in the stochastic direction and allows for spatially heterogeneous stochastic structure. This new concept of sparsity motivates a heterogeneous stochastic finite element method (**HSFEM**) framework for linear elliptic equations, which discretizes the equations using the heterogeneous coupling of spatial basis with local stochastic basis to exploit the local stochastic structure of the solution space. We also provide a sampling method to construct the local stochastic basis for this framework using the randomized range finding techniques. The resulting HSFEM involves two stages and suits the multi-query setting: in the offline stage, the local stochastic structure of the solution space is identified; in the online stage, the equation can be efficiently solved for multiple forcing functions. An online error estimation and correction procedure through Monte Carlo sampling is given. Numerical results for several problems with high dimensional stochastic input are presented to demonstrate the efficiency of the HSFEM in the online stage.

1 Introduction and Main Results

Analysis of complex systems requires not only a fine understanding of the underlying physics, but also recognition of the intrinsic uncertainties and their influence on the quantities of interest. Uncertainty Quantification (**UQ**) is an emerging discipline that aims at addressing the latter issue and has attracted growing interest recently. In this paper we consider UQ of the following linear elliptic equation with stochastic coefficient, which can be used to model diffusion processes in random media:

$$\begin{cases} -\operatorname{div}(a(x, \omega) \nabla u(x, \omega)) = f(x), & x \in D, \quad \omega \in \Omega, \\ u(x, \omega)|_{\partial D} = 0. \end{cases} \quad (1.1)$$

Here D is a bounded convex polygon domain in R^d , and (Ω, \mathcal{F}, P) is a probability space with $\Omega \subset R^m$, i.e., the dimension of the stochastic input ω is m . We assume that $f(x) \in L^2(D)$, and $a(x, \omega)$ is bounded and uniformly elliptic, i.e., there exist λ_{\min} and λ_{\max} such that

$$P(\omega \in \Omega : a(x, \omega) \in [\lambda_{\min}, \lambda_{\max}], \quad \forall x \in D) = 1. \quad (1.2)$$

The solution to equation (1.1) is $u(x, \omega) \in L^2(H_0^1(D), \Omega)$, such that for any $\phi(x, \omega) \in L^2(H_0^1(D), \Omega)$,

$$\int_{\Omega} \int_D \nabla u(x, \omega)^T a(x, \omega) \nabla \phi(x, \omega) dx dP = \int_{\Omega} \int_D f(x) \phi(x, \omega) dx dP, \quad (1.3)$$

where the function space $L^2(H_0^1(D), \Omega)$ is a Hilbert space with inner product defined as

$$\langle u(x, \omega), v(x, \omega) \rangle = \int_{\Omega} \int_D \nabla u(x, \omega)^T \nabla v(x, \omega) dx dP. \quad (1.4)$$

The existence of solution to equation (1.1) can be obtained using the Lax-Milgram Theorem,

$$\|u(x, \omega)\|_{L^2(H_0^1(D), \Omega)} \leq C \|f(x)\|_{H^{-1}(D)}, \quad (1.5)$$

and more theoretical aspects of this equation can be found in [3].

Several types of numerical methods have been proposed for UQ of Stochastic Partial Differential Equations (**SPDE**). Perturbation Methods (**PM**) [15, 1, 22] start with expanding the stochastic solution via Taylor expansion and result in a system of deterministic equations by truncating after certain

*hou@cms.caltech.edu, Computing and Mathematical Sciences, CalTech, Pasadena, CA 91125.

†Correspondence. plliu@caltech.edu, Computing and Mathematical Sciences, CalTech, Pasadena, CA 91125.

order. Typically at most second-order expansions are used as the system of equations becomes very cumbersome if higher-order terms are included. One limitation of perturbation methods is that the magnitude of the input and output uncertainty must be small compared with their respective means. Monte Carlo Methods (**MC**) [7, 27] remain popular for SPDE problems because of its non-intrusive nature especially when the dimension of the stochastic input is high. According to the law of large numbers, the convergence rate of MC is only $O(M^{-1/2})$, where M is the number of realizations. This low convergence rate limits its application. MC Methods are also sensitive to the random number generator and typically produce output with uncertain accuracy. Stochastic Collocation Methods (**SC**) [2, 28, 31, 26, 36] collocate the problem in zeros of tensor product orthogonal polynomials and the solution is recovered using interpolation. SC methods attain high accuracy when the solution is smooth with respect to the random variables. However, a very large number of collocation points are required to obtain accuracy when the stochastic input has high dimension because of the tensor product. This is called the *curse of dimensionality*. Polynomial Chaos Methods (**PC**) [35, 8, 20, 37, 14] project the solution $u(x, \omega)$ to an orthogonal polynomial basis $H_\alpha(\omega)$ with respect to the underlying probability distribution, and approximate the solution by $u(x, \omega) \approx \sum_\alpha u_\alpha(x) H_\alpha(\omega)$. The coefficients $u_\alpha(x)$ can be obtained by solving a system of coupled elliptic equations. PC Methods also suffer from the *curse of dimensionality* because a large number of polynomial basis functions are required when the stochastic dimension is high.

SPDE problems with high stochastic dimension are very challenging because of the *curse of dimensionality*. There have been many attempts in the literature, e.g. [9, 10, 11, 12, 13, 23, 30, 6, 5, 4] to attack these challenging problems. Most of them take advantage of the fact that even though the stochastic input has high dimension, the solution actually lives in a relatively low dimensional space, i.e., enjoys some sense of sparsity. In [13, 23, 30] the compressive sensing technique is employed to identify a sparse representation of the solution in the stochastic direction. In [10, 11, 12], the Karhunen-Loève expansion of the solution is used to compactly represent the solution and reduce degrees of freedom. In [21, 29], the authors construct a reduced spatial basis using snapshots of the solutions to obtain computational savings for each realization of the stochastic equation. The present work also seeks to attack the *curse of dimensionality* by exploring the sparsity of the solution space. Our methodology differs from the previous sparsity-exploiting methods in the following two aspects: we use different stochastic basis functions in different regions of the domain to approximate the solution, allowing for spatially heterogeneous stochastic structure of the solution space; we seek a sparse representation of the whole solution space for all $f(x) \in L^2(D)$, not a specific solution, thus our method suits the multi-query setting.

In this work we first introduce a new concept of sparsity for the stochastic operator $-\text{div}(a(x, \omega)\nabla(\cdot))$. We consider a finite dimensional approximation to the solution space of (1.1) taking the form of

$$V_h = \left\{ \sum_{i=1}^n \sum_{j=0}^{k_i} c_i^j \phi_i(x) \xi_i^j(\omega) : c_i^j \in R \right\}, \quad (1.6)$$

where $\{\phi_i(x)\}$ is a standard piecewise linear basis that can resolve the spatial variation of the solution, and $\xi_i^j(\omega)$, $j = 0, \dots, k_i$ are the **local stochastic basis** functions associated with $\phi_i(x)$. Note that in (1.6), different local stochastic basis functions are used in different regions of the domain to approximate the solution, and this allows for spatially heterogeneous stochastic structure of the solution space to (1.1). To obtain certain accuracy using (1.6), the required $k = \sum_{i=1}^n k_i/n$, which is the relative size of V_h to the approximate solution space of the corresponding deterministic equation, measures the compactness of the inverse of the stochastic operator in the stochastic direction. If k is *small*, then we say the stochastic operator enjoys the Operator-Sparsity. We call it **weak** or **strong** Operator-Sparsity depending on whether the approximation is taken to be in $L^2(D \times \Omega)$ or $L^2(H_0^1(D), \Omega)$. An interpretation of this Operator-Sparsity in terms of the decay rate of the singular values in the Karhunen-Loève expansion of the Green's function is given. We also prove that to obtain certain approximation accuracy using V_h in $L^2(D \times \Omega)$, the required k has an upper bound that only depends on the ellipticity of the operator, λ_{\min} and λ_{\max} , and in particular, does not depend on the dimension of the stochastic input.

This new concept of sparsity motivates a heterogeneous stochastic finite element method framework (**HSFEM**) for linear elliptic equations. We first construct a local stochastic basis $\{\xi_i^j(\omega)\}$ and trial space V_h (1.6). Then based on (1.3), we define the numerical solution as $u_h(x, \omega) \in V_h$, such that

$$\int_{\Omega} \int_D \nabla u_h(x, \omega)^T a(x, \omega) \nabla v(x, \omega) dx dP = \int_{\Omega} \int_D f(x) v(x, \omega) dx dP, \quad \forall v(x, \omega) \in V_h.$$

The numerical solution defined as above satisfies the following quasi-optimality property

$$\|u(x, \omega) - u_h(x, \omega)\|_{L^2(H_0^1(D), \Omega)} \leq \lambda_{\max}/\lambda_{\min} \inf_{v \in V_h} \|u(x, \omega) - v(x, \omega)\|_{L^2(H_0^1(D), \Omega)}.$$

The key difference of this HSFEM framework from classical methods like PC or SC methods is that different local stochastic basis functions are used in different regions of the domain to discretize the equation, which allows for spatially heterogeneous stochastic structure of the solution space. If the operator enjoys the (strong) Operator-Sparsity, this HSFEM framework can attain high accuracy using only a small trial space V_h . This HSFEM framework can be viewed as a generalization of the PC methods using **problem-dependent** and **local** stochastic basis, and provides a novel direction to attack the *curse of dimensionality* by exploiting the local stochastic structure of the solution space.

We also provide a sampling method to construct the local stochastic basis for the HSFEM framework using the randomized range finding techniques. Given the HSFEM framework, a suitable local stochastic basis $\{\xi_i^j(\omega)\}$ still need to be constructed. Since the spatial basis $\phi_i(x)$ has local support around the node point x_i , the local stochastic basis functions $\xi_i^j(\omega), j = 0, \dots, k_i$ in the trial space (1.6) are only used to approximate the solution in the stochastic direction near x_i . So we consider the following linear operator which maps the forcing function to the stochastic part of the solution at x_i ,

$$T_i : L^2(D) \rightarrow L^2(\Omega), \quad f(x) \rightarrow u(x_i, \omega) - E[u(x_i, \omega)].$$

T_i can be shown to be a compact linear operator thus can be approximated by a matrix. We apply the randomized range finding techniques [19] to T_i to construct the local stochastic basis functions $\xi_i^j(\omega), j = 1, \dots, k_i$. The resulting local stochastic basis functions are close to optimal in approximating the range of T_i in the $L^2(\Omega)$ sense if the singular values of T_i decay fast.

The HSFEM framework combined with the sampling method to construct the local stochastic basis constitutes a heterogeneous stochastic finite element method. This method involves two stages:

- In the offline stage we identify the sparse structure of the solution space. We construct the local stochastic basis by solving the SPDE for a number of randomly chosen forcing functions followed by some orthogonalization process, and then build the corresponding stiffness matrix.
- In the online stage, the equation can be efficiently solved for multiple forcing functions using the stiffness matrix constructed in the offline stage, which is small and sparse.

The offline computation of the HSFEM is expensive since it involves solving the SPDE for a number of times using traditional methods, and this limits its application to the multi-query setting, which means the equation needs to be solved multiple times using different forcing functions. Our method suits the multi-query setting because the linear system we solve online is sparse and small. Methods to reduce the offline computation cost and an online error estimation and correction procedure are given.

We present numerical results for several problems with high dimensional stochastic input to demonstrate the efficiency of the HSFEM. In our numerical examples, we show that the solution space to stochastic elliptic equation has spatially heterogeneous stochastic structure, and this heterogeneity can be recognized by the HSFEM in the offline stage. Moreover, the number of the constructed local stochastic basis functions is small and the online numerical solutions have high accuracy, which implies that the stochastic operators that we consider enjoy the Operator-Sparsity and the local stochastic basis constructed using the sampling method works well within the HSFEM framework. We also compare the accuracy of the HSFEM numerical solution with the truncated Karhunen-Loève expansion of the solution to demonstrate the advantage of approximating the stochastic part of the solution locally.

The rest part of this paper is organized as follows. In section 2, we introduce the Karhunen-Loève expansion and the new concept of sparsity. In section 3, we develop the HSFEM framework. In section 4, we discuss the sampling method to construct the local stochastic basis. In section 5, we address several issues related to the numerical implementation of the HSFEM. In section 6, we present our numerical results. Section 7 is devoted to concluding remarks and future work.

2 A New Concept of Sparsity for Stochastic Elliptic Operator

In this section, we first introduce the Karhunen-Loève (KL) expansion [15, 32] and the sense of sparsity related to the KL expansion. Two observations are made on the KL expansion, which motivate an Operator-Sparsity allowing for spatially heterogeneous stochastic structure of the corresponding solution space. An interpretation of this new concept of sparsity in terms of the Green's function is given.

2.1 Karhunen-Loève Expansion

If a stochastic process $u(x, \omega) \in L^2(D \times \Omega)$, then it can be expanded in a Fourier-type series as:[15]

$$u(x, \omega) = \bar{u}(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \psi_i(x) \xi_i(\omega), \quad \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_i \geq \dots > 0, \quad (2.1)$$

where $\bar{u}(x) = \int_{\Omega} u(x, \omega) dP$, $\psi_i(x)$ are orthonormal in $L^2(D)$ and $\xi_i(\omega)$ are uncorrelated uni-variate random variables with mean 0. (2.1) is called the Karhunen-Loève (**KL**) expansion of $u(x, \omega)$, and $\psi_i(x)$ can be computed as eigenfunctions of the covariance function of $u(x, \omega)$. The k -term truncation

$$\sum_{i=1}^k \sqrt{\lambda_i} \psi_i(x) \xi_i(\omega) \quad (2.2)$$

is the best rank- k approximation of $u(x, \omega) - \bar{u}(x)$ in $L^2(D \times \Omega)$. [15]

If the singular values $\sqrt{\lambda_i}$ decay very fast in the KL expansion, then the solution has data-sparsity in the sense that a few data can provide an accurate description of the solution. More discussion about the data-sparsity can be found in [17, 18]. This sense of sparsity has been exploited to reduce the computation cost in [9, 10, 11, 12], and we refer it as the KL-sense Sparsity.

Definition 2.1 (KL-sense Sparsity). The solution $u(x, \omega)$ to equation (1.1) is said to have KL-sense sparsity if the singular values $\sqrt{\lambda_i}$ in its KL expansion decay very fast (e.g. exponential decay), and a small number of modes are sufficient to represent the solution accurately.

Remark 2.1. Here we leave the notions ‘decay very fast’ and ‘accurately’ vague, because they depend on the nature of the problem and the desired order of accuracy.

We make the following two observations on the KL expansion and the KL-sense sparsity:

- The KL-sense sparsity reflects the property of some solution to equation (1.1) with a specific forcing, not the property of the solution space. The first several stochastic basis functions $\xi_i(\omega)$ in the KL expansion of one specific solution do not necessarily approximate another solution well.
- The truncated KL expansion (2.2) seeks to approximate the stochastic behavior of the solution in different regions of the domain using the same set of random variables $\xi_i(\omega), i = 1, \dots, k$, which is quite restrictive since the solution may have spatially heterogeneous stochastic structure.

These two observations motivate us to define the following Operator-Sparsity.

2.2 A New Concept of Sparsity for the Stochastic Operator

We consider the inverse of the stochastic elliptic operator $L(x, \omega) = -\text{div}(a(x, \omega) \nabla(\cdot))$, which is denoted as $L^{-1}(x, \omega)$. Then $L^{-1}(x, \omega)$ maps $f(x) \in L^2(D)$ to $u(x, \omega)$, which is the solution to equation (1.1).

Remark 2.2 (Random Forcing Functions). In this paper, we only consider deterministic forcing $f(x)$. For the case that the right hand side of (1.1) has randomness which we denote by $f(x, \omega)$, we can first decompose $f(x, \omega)$ (for example, using KL expansion) to

$$f(x, \omega) \approx f_0(x) + \sum_{i=1}^{N_f} f_i(x) \xi_i(\omega).$$

Then based on (1.1), the corresponding solution can be approximated using

$$u(x, \omega) \approx L^{-1}(x, \omega) f_0(x) + \sum_{i=1}^{N_f} \xi_i(\omega) L^{-1}(x, \omega) f_i(x),$$

which means once we have obtained an efficient solver for equation (1.1) with deterministic forcing, we can also solve the equation with random forcing efficiently.

We seek to construct a finite dimensional linear operator $L_h^{-1}(x, \omega)$ to approximate $L^{-1}(x, \omega)$. It is known that the Lagrange basis (piecewise polynomials) can approximate the solution to deterministic

elliptic equation very well [33] if the coefficient $a(x)$ is in $C^1(D)$. For the SPDE, it is natural to use the product (**not** tensor product) of these Lagrange basis functions with some stochastic basis to approximate the solution to equation (1.1). To be specific: assume D is equipped with a triangular mesh with node points $x_i, i = 1, \dots, n$, and $\phi_i(x)$ are the piecewise linear functions on D satisfying

$$\phi_i(x_j) = \delta_{ij}.$$

Then we want to construct a stochastic basis $\{\xi_i^j(\omega)\}$, and use

$$V_h = \left\{ \sum_{i=1}^n \sum_{j=0}^{k_i} c_i^j \phi_i(x) \xi_i^j(\omega) : c_i^j \in R \right\} \quad (2.3)$$

to approximate the solution space of equation (1.1). Note that V_h consists of the heterogeneous coupling of spatial basis with stochastic basis, namely, we use different stochastic basis functions to couple with different spatial basis functions to approximate the solution space to (1.1), and this allows for spatially heterogeneous stochastic structure of the solution space. We want

$$\sup_{f(x) \in L^2(D)} \frac{\inf_{c_i^j \in R} \|u(x, \omega) - \sum_{i=1}^n \sum_{j=0}^{k_i} c_i^j \phi_i(x) \xi_i^j(\omega)\|_H}{\|f(x)\|_{L^2(D)}} \leq \epsilon, \quad (2.4)$$

where $\|\cdot\|_H$ can be taken to be the $L^2(H_0^1(D), \Omega)$ norm or $L^2(D \times \Omega)$ norm.

We call $\{\phi_i(x) \xi_i^j(\omega)\}$ the **Coupling Basis**. Assume that to obtain **certain order of accuracy**, the total number of the coupling basis functions in the above approximation (2.4) is

$$S = \sum_{i=1}^n (k_i + 1).$$

The ratio of S over n is the relative size of the approximate solution space of (1.1), V_h , to the corresponding deterministic equation, thus measures the compactness of the inverse operator $L^{-1}(x, \omega)$ in the stochastic direction. This motivates us to define the following sense of sparsity.

Definition 2.2 (Operator-Sparsity). For the stochastic elliptic operator

$$L(x, \omega) = -\operatorname{div}(a(x, \omega) \nabla(\cdot)),$$

if the inverse of $L(x, \omega)$ can be well-approximated using finite dimensional operator in the sense of (2.4), and the average number of stochastic basis functions for each node point is small, i.e.,

$$k = \frac{1}{n} \sum_{i=1}^n k_i \quad \text{is small,}$$

then we say that $L(x, \omega)$ has Operator-Sparsity.

If the approximation is taken to be in the $L^2(H_0^1(D), \Omega)$ sense, we call it the **strong** Operator-Sparsity. Or if the approximation is taken to be in the $L^2(D \times \Omega)$ sense, we call it the **weak** Operator-Sparsity.

Remark 2.3. We leave the notions ‘well-approximated’ and ‘small’ vague since they depend on the desired order of accuracy and the nature of the problem.

2.3 Interpretation of the Operator-Sparsity

Since $\phi_i(x)$ has compact support near x_i , the $\xi_i^j(\omega)$, $0 = 1, \dots, k_i$, in (2.4) are only used to approximate the solution in the stochastic direction near x_i . So we call $\{\xi_i^j(\omega)\}$ the **Local Stochastic Basis**.

To give an interpretation of this new concept of sparsity, we consider the local stochastic behavior of the solution near x_i . Let $G(x, y, \omega)$ be the Green’s function corresponding to the elliptic operator $-\operatorname{div}(a(x, \omega) \nabla(\cdot))$, then the solution to equation (1.1) at x_i is

$$u(x_i, \omega) = \int_D G(x_i, y, \omega) f(y) dy.$$

For $d \leq 3$, based on the estimates of Green’s function in [16, 34], we have that for fixed x_i and $\omega \in \Omega$,

$$\|G(x_i, y, \omega)\|_{L^2(D)} \leq C_i, \quad (2.5)$$

where C_i depends on x_i , λ_{\min} , λ_{\max} and the domain D , but does not depend on the dimension of ω . Integrating (2.5) in Ω , we get

$$\|G(x_i, y, \omega)\|_{L^2(D \times \Omega)} \leq C_i < +\infty.$$

So the Green's function $G(x_i, y, \omega)$ has KL expansion

$$G(x_i, y, \omega) = \bar{G}(x_i, y) + \sum_{j=1}^{\infty} \sigma_i^j \psi_i^j(y) \eta_i^j(\omega), \quad \sum_{j=1}^{\infty} (\sigma_i^j)^2 \leq C_i^2, \quad (2.6)$$

where $\psi_i^j(y)$ and $\eta_i^j(\omega)$, $j = 1, \dots, \infty$ are orthonormal in $L^2(D)$ and $L^2(\omega)$ respectively. Then

$$u(x_i, \omega) - \bar{u}(x_i) = \sum_{j=1}^{\infty} \sigma_i^j \eta_i^j(\omega) \int_D f(y) \psi_i^j(y) dy. \quad (2.7)$$

Denote k_i as the number of required terms to make the singular values σ_i^j decay to ϵ , i.e.,

$$k_i = \min\{j : \sigma_i^j < \epsilon\}. \quad (2.8)$$

Since $\sum_{j=1}^{\infty} (\sigma_i^j)^2 \leq C_i^2$, we have

$$k_i \leq [C_i^2 / \epsilon^2], \quad (2.9)$$

where the right hand side does not depend on the dimension of the stochastic input ω .

When the first k_i stochastic basis functions are used to approximate the stochastic solution, we have

$$\|u(x_i, \omega) - \bar{u}(x_i) - \sum_{j=1}^{k_i} \sigma_i^j \eta_i^j(\omega) \int_D f(y) \psi_i^j(y) dy\|_{L^2(\Omega)} \leq \sigma_i^{k_i+1} \|f(x)\|_{L^2(\Omega)} \leq \epsilon \|f(x)\|_{L^2(D)}. \quad (2.10)$$

Remark 2.4. The estimate (2.9) corresponds to the worst case that the first $[C_i^2 / \epsilon^2]$ singular values in the KL expansion of $G(x_i, y, \omega)$ are equal to ϵ and others are 0. It is a very pessimistic estimate, and according to our numerical results in section 6.2.1, the singular values σ_i^j actually decay exponentially fast (for that particular example), which means there exist $c_i, C_i > 0$, such that

$$\sigma_i^j \leq C_i e^{-c_i j}.$$

Then to obtain ϵ -accuracy in approximating the stochastic part of the solution at x_i , only

$$k_i = \lceil \frac{\log C_i - \log \epsilon}{c_i} \rceil, \quad (2.11)$$

stochastic basis functions are needed. The decay rate of the singular values in the KL expansion of the Green's function $G(x_i, y, \omega)$ will be investigated in another work.

So if the singular values in the KL expansion of $G(x_i, y, \omega)$, σ_i^j , decay fast, a small number of stochastic basis functions $\eta_i^j(\omega)$, $j = 1, \dots, k_i$ can approximate the local stochastic behavior of the solution near x_i well. This is very close to (but not exactly the same as) our definition of Operator-Sparsity, so we can interpret the Operator-Sparsity as the **low-rankness** of the Green's function $G(x, y, \omega)$ for fixed x .

To make this interpretation rigorous, we denote $Ju(x, \omega)$ as the piecewise linear interpolation of the solution $u(x, \omega)$ using the spatial basis $\phi_i(x)$, which means $Ju(x_i, \omega) = u(x_i, \omega)$. We assume that the spatial basis $\phi_i(x)$, $i = 1, \dots, n$ can resolve the variation of the solution in the spatial direction, then

$$\|u(x, \omega) - Ju(x, \omega)\|_{L^2(D \times \Omega)} \quad (2.12)$$

is small and we ignore this interpolation error. We assume that the domain is divided into elements τ_i , $i = 1, \dots, n_t$, which are triangles for $d = 2$ and tetrahedrons for $d = 3$, and let x_i^j be the nodes of τ_i .

Lemma 2.1. Choose the first k_i (2.8) stochastic basis functions in the KL expansion of $G(x_i, y, \omega)$, $\eta_i^j(\omega)$, $j = 1, \dots, k_i$ as the local stochastic basis in (2.3) and let $\xi_i^0(\omega) = 1$. Then

$$\inf_{v(x, \omega) \in V_h} \|u(x, \omega) - v(x, \omega)\|_{L^2(D \times \Omega)} \leq C \epsilon \|f(x)\|_{L^2(D)}. \quad (2.13)$$

Proof. Let $c_i^0 = \bar{u}(x_i)$, $c_i^j = \sigma_i^j \int_D f(y) \psi_i^j(y) dy$ for $j > 0$, and denote $Pu(x, \omega) \in V_h$ as

$$Pu(x, \omega) = - \sum_{i=1}^n \sum_{j=0}^{k_i} c_i^j \phi_i(x) \eta_i^j(\omega).$$

Then

$$Ju(x_i, \omega) - Pu(x_i, \omega) = u(x_i, \omega) - \bar{u}(x_i) - \sum_{j=1}^{k_i} \sigma_i^j \eta_i^j(\omega) \int_D f(y) \psi_i^j(y) dy.$$

According to (2.10),

$$\|Ju(x_i, \omega) - Pu(x_i, \omega)\|_{L^2(\Omega)} \leq \epsilon \|f(x)\|_{L^2(D)}. \quad (2.14)$$

Consider the $L^2(D \times \Omega)$ norm of the error,

$$\|Ju(x, \omega) - Pu(x, \omega)\|_{L^2(D \times \Omega)}^2 = \int_{\Omega} \|Ju(x, \omega) - Pu(x, \omega)\|_{L^2(D)}^2 dP. \quad (2.15)$$

Since for fixed ω , $Ju(x, \omega) - Pu(x, \omega)$ is piecewise linear, we have

$$\|Ju(x, \omega) - Pu(x, \omega)\|_{L^2(D)}^2 = \sum_{i=1}^{n_t} \|Ju(x, \omega) - Pu(x, \omega)\|_{L^2(\tau_i)}^2 \quad (2.16)$$

$$\leq \sum_{i=1}^{n_t} |\tau_i| \left(\sum_j (Ju(x_i^j, \omega) - Pu(x_i^j, \omega))^2 \right). \quad (2.17)$$

Putting the estimate (2.15) in (2.17), we get

$$\|Ju(x, \omega) - Pu(x, \omega)\|_{L^2(D \times \Omega)}^2 \leq \sum_{i=1}^{n_t} |\tau_i| \sum_j \|Ju(x_i^j, \omega) - Pu(x_i^j, \omega)\|_{L^2(\Omega)}^2.$$

According to (2.14), we get

$$\|Ju(x, \omega) - Pu(x, \omega)\|_{L^2(D \times \Omega)} \leq C\epsilon \|f(x)\|_{L^2(D)}.$$

With (2.12), we complete the proof. \square

Remark 2.5. To obtain a similar estimate for the approximation error of V_h in the $L^2(H_0^1(D), \Omega)$ sense, we need the regularity of the error in the spatial direction. We leave this issue to our future work.

Based on (2.9) and estimate (2.13), we have that to obtain certain approximation accuracy of the solution space to (1.1) using finite dimensional space (2.3) in $L^2(D \times \Omega)$, the required number of local stochastic basis functions k_i has an upper bound that only depends on the ellipticity of the stochastic operator, λ_{\min} and λ_{\max} , and in particular, does not depend on the dimension of the stochastic input.

Before we go to the next section, we make the following observations on this Operator-Sparsity.

- The Operator-Sparsity reflects the compactness of $L^{-1}(x, \omega)$ in the stochastic direction, but it is **stronger** than compactness because we have restricted the finite dimensional space approximating the range of $L^{-1}(x, \omega)$ to take the form of (2.3).
- The Operator-Sparsity permits using different stochastic basis functions in different regions of the domain to approximate the solution thus allows for spatially heterogeneous stochastic structure of the solution space. In this sense it is **weaker** than the KL-sense Sparsity.
- The Operator-Sparsity reflects the property of the stochastic elliptic operator $-\operatorname{div}(a(x, \omega) \nabla(\cdot))$ independent of the forcing function, thus it can be naturally exploited to solve equation (1.1) for multiple forcing functions and significantly reduce the computation cost.

Our definition of Operator-Sparsity (2.4) together with the weak formulation of the stochastic elliptic equation (1.3) leads to a heterogeneous stochastic finite element method framework (**HSFEM**) to solve linear stochastic elliptic equations, which is discussed in detail in the next section.

3 A Heterogeneous Stochastic FEM Framework

Define a functional J on $L^2(H_0^1(D), \Omega)$ as

$$J(u(x, \omega)) = \frac{1}{2} \int_{\Omega} \int_D \nabla u(x, \omega)^T a(x, \omega) \nabla u(x, \omega) dx dP - \int_{\Omega} \int_D u(x, \omega) f(x) dx dP.$$

Then we have

Theorem 3.1 (Variational Formulation). *The weak formulation of the stochastic elliptic equation (1.3) is equivalent to finding*

$$u(x, \omega) \in L^2(H_0^1(D), \Omega),$$

such that

$$J(u(x, \omega)) \leq J(v(x, \omega)), \quad \forall v(x, \omega) \in L^2(H_0^1(D), \Omega). \quad (3.1)$$

Based on (3.1), given a finite dimensional subspace of $L^2(H_0^1(D), \Omega)$, V_h , the corresponding finite element formulation of equation (1.1) is finding $u_h(x, \omega) \in V_h$ such that

$$J(u_h(x, \omega)) \leq J(v(x, \omega)), \quad \forall v(x, \omega) \in V_h. \quad (3.2)$$

Theorem 3.2 (Finite Element Formulation). *The finite element formulation (3.2) is equivalent to finding*

$$u_h(x, \omega) \in V_h,$$

such that

$$\int_{\Omega} \int_D \nabla u_h(x, \omega)^T a(x, \omega) \nabla v(x, \omega) dx dP = \int_{\Omega} \int_D f(x) v(x, \omega) dx dP, \quad \forall v \in V_h. \quad (3.3)$$

The numerical solution defined in this way has the following property.

Theorem 3.3 (Quasi-Optimality).

$$\|u(x, \omega) - u_h(x, \omega)\|_{L^2(H_0^1(D), \Omega)} \leq \lambda_{\max}/\lambda_{\min} \inf_{v \in V_h} \|u(x, \omega) - v(x, \omega)\|_{L^2(H_0^1(D), \Omega)}. \quad (3.4)$$

If we choose the finite dimensional subspace of $L^2(H_0^1(D), \Omega)$ to be of the form (2.3), i.e., the trial space V_h is chosen to be the heterogeneous coupling of spatial basis with local stochastic basis, then we get the following HSFEM framework for solving linear stochastic elliptic equations.

The Heterogeneous Finite Element Method Framework

- Construct the local stochastic basis $\xi_i^j(\omega)$, $i = 1, \dots, n$, $j = 0, \dots, k_i$, which can capture the stochastic behavior of the solution near x_i . Let $\xi_i^0(\omega) = 1$.
- Find $c_i^j, i = 1, \dots, n, j = 0, \dots, k_i$, such that for all $i' = 1, \dots, n, j' = 0, \dots, k_{i'}$,

$$\int_{\Omega} \int_D \sum_{i=1}^n \sum_{j=0}^{k_i} c_i^j \xi_i^j \nabla \phi_i(x)^T a(x, \omega) \xi_{i'}^{j'} \nabla \phi_{i'}(x) dx dP = \int_{\Omega} \int_D f(x) \xi_{i'}^{j'} \phi_{i'}(x) dx dP.$$

The corresponding linear system is uniquely solvable if for each i , $\xi_i^j(\omega)$, $j = 0, \dots, k_i$ are linearly independent, i.e.,

$$\sum_{j=0}^{k_i} c_j \xi_i^j(\omega) = 0 \quad \Leftrightarrow \quad c_j = 0.$$

- Recover the numerical solution as

$$u_h(x, \omega) = \sum_{i=1}^n \sum_{j=0}^{k_i} c_i^j \phi_i(x) \xi_i^j(\omega).$$

- *A posteriori* error estimation and correction.

The novelty of the HSFEM framework is that the equation is discretized in the stochastic and spatial directions simultaneously using the heterogeneous coupling of spatial basis with local stochastic basis, and this allows for spatially heterogeneous stochastic structure of the solution space. Given this HSFEM framework, a suitable local stochastic basis still need to be chosen. Different choices of $\{\xi_i^j(\omega)\}$ lead to different numerical formulations of equation (1.1). For example, if $\xi_i^j(\omega)$ are chosen to be the orthonormal polynomials $H_\alpha(\omega)$ in the PC methods for all i , then the HSFEM is equivalent to the PC [3, 37] methods. This implies that the HSFEM framework is a generalization of the PC methods using **problem-dependent** and **local** stochastic basis to approximate the solution space. If the stochastic operator enjoys the (strong) Operator-Sparsity and a suitable local stochastic basis $\{\xi_i^j(\omega)\}$ is chosen, then based on (2.4) and (3.4), this HSFEM framework can attain high accuracy using only a small number of coupling basis functions $S = \sum_i^n k_i + n$. So the HSFEM framework provides a novel direction to attack the *curse of dimensionality* by exploiting local stochastic structure of the solution space.

In the following section, we provide a sampling method to construct the local stochastic basis $\{\xi_i^j(\omega)\}$ using the randomized range finding techniques [19]. The constructed local stochastic basis functions are close to optimal in capturing the local stochastic behavior of the solution space in the $L^2(\Omega)$ sense if the singular values in the KL expansion of the Green's function decay fast.

4 A Sampling Method to Construct the Local Stochastic Basis

In this section we first introduce the compact linear operator T_i , which maps the forcing $f(x)$ to the stochastic part of the solution at node point x_i . We apply the randomized range finding techniques to T_i to construct the local stochastic basis. Numerical implementation details are given.

4.1 The Linear Compact Operator T_i

Since the local stochastic basis functions $\xi_i^j(\omega), j = 1, \dots, k_i$ are used to approximate the behavior of the solution in the stochastic direction near x_i , we consider the following linear operator that maps the forcing function to the stochastic part of the solution at x_i :

$$T_i : f(x) \mapsto u(x_i, \omega) - E[u(x_i, \omega)]. \quad (4.1)$$

In dimension $d \leq 3$, T_i is actually a compact linear operator mapping from $L^2(D)$ to $L^2(\Omega)$, whose singular value decomposition has already been given by (2.7),

$$T_i f(y) = \sum_{j=1}^{\infty} \sigma_i^j \eta_i^j(\omega) \int_D \psi_i^j(y) f(y) dy,$$

where $\sigma_i^j, \eta_i^j(\omega)$ and $\psi_i^j(x)$ come from the KL expansion of the Green's function (2.6). The subspace of $L^2(\Omega)$ spanned by the first k left singular vectors of T_i

$$\text{span}\{\eta_i^1(\omega), \eta_i^2(\omega), \dots, \eta_i^k(\omega)\} \quad (4.2)$$

is the best k dimensional approximation of the range of T_i in the $L^2(\Omega)$ sense. So $\eta_i^j(\omega), j = 1, \dots, k_i$, are good candidates for the local stochastic basis functions $\xi_i^j(\omega), j = 1, \dots, k_i$.

Based on our analysis of the Green's function in section 2.3, we have $\sum_{j=1}^{\infty} (\sigma_i^j)^2 < C_i^2$. So to obtain ϵ accuracy in approximating T_i in the $L^2(\Omega)$ sense, at most $\lceil C_i^2/\epsilon^2 \rceil$ (2.9) stochastic basis functions are required, which depends on the ellipticity of the operator, but is independent of the dimension of ω .

However, it is impractical to construct $\text{span}\{\eta_i^1(\omega), \eta_i^2(\omega), \dots, \eta_i^k(\omega)\}$ by doing KL expansion to the Green's function. In this section we develop a sampling method using the randomized range finding algorithm to construct the local stochastic basis, which can capture the main action of T_i and is an approximation of (4.2). The basic idea is using T_i to act on some random matrix Ω , then the main action of T_i can be extracted through an orthogonalization process from the image $T_i \Omega$.

4.2 Discretization of T_i

We first discretize the domain of $T_i, L^2(D)$ using a set of orthonormal basis functions

$$\hat{D} = \{\Phi_1(x), \Phi_2(x), \dots, \Phi_N(x)\}. \quad (4.3)$$

In our numerical examples in section 6, the domain D is chosen to be $[0, 1]^d$, and we discretize D using uniform mesh of size h . \hat{D} is chosen to be the first $N = (2l + 1)^d$ Fourier modes with $l = \frac{1}{2h}$, i.e.,

$$\hat{D} = \otimes_{i=1}^d \{1, \dots, 2 \sin(2\pi l x_i), 2 \cos(2\pi l x_i)\}, \quad (4.4)$$

which contains all the Fourier modes that can be resolved by the given mesh.

Remark 4.1. $L^2(D)$ is an infinite dimensional space and cannot be approximated using finite basis functions. However, in this paper we seek to construct finite dimensional space V_h (2.3) to approximate the solution space to (1.1). Assuming now the constructed space V_h can make (2.4) hold for all $f(x) \in \hat{D}$, i.e., for any $f(x) \in \hat{D}$, there exists $u_h(x, \omega) \in V_h$, such that

$$\|u(x, \omega) - u_h(x, \omega)\|_H \leq \epsilon \|f(x)\|_{L^2(D)}.$$

where $\|\cdot\|_H$ can be either $L^2(D \times \Omega)$ or $L^2(H_0^1(D), \Omega)$ norm. Then for any $f(x) \in L^2(D)$, we have $f(x) = f^1(x) + f^2(x)$, where $f^1(x) \in \hat{D}$ and $f^2(x) \in \hat{D}^\perp$. \hat{D}^\perp is the orthogonal complement of \hat{D} in $L^2(D)$ and consists of high frequency Fourier modes. Then for $f^2(x) \in \hat{D}^\perp$, $\|f^2(x)\|_{H^{-1}(D)} \leq C/l \|f^2(x)\|_{L^2(D)} \leq Ch \|f^2(x)\|_{L^2(D)}$ according to (4.4) and the fact that we have chosen $l = 1/2h$.

The solution $u(x, \omega)$ can be divided into $u(x, \omega) = u^1(x, \omega) + u^2(x, \omega)$ correspondingly, and there exists $u_h^1(x, \omega) \in V_h$ such that $\|u^1(x, \omega) - u_h^1(x, \omega)\|_H \leq \epsilon \|f^1(x)\|_{L^2(D)}$. Then

$$\|u(x, \omega) - u_h^1(x, \omega)\|_H \leq \epsilon \|f^1(x)\|_{L^2(D)} + \|u^2(x, \omega)\|_H,$$

and according to (1.5), $\|u^2(x, \omega)\|_H \leq C \|f^2(x)\|_{H^{-1}(D)} \leq Ch \|f^2(x)\|_{L^2(D)}$. Then

$$\|u(x, \omega) - u_h^1(x, \omega)\|_H \leq (\epsilon + Ch) \|f(x)\|_{L^2(D)} \leq C\epsilon \|f(x)\|_{L^2(D)}, \quad (4.5)$$

where we have assumed h is small ($h \leq \epsilon$) in the second inequality. (4.5) implies that when constructing trial space V_h to approximate the solution space of (1.1) for $f(x) \in L^2(D)$, we do not need to consider the high frequency part of $L^2(D)$ that cannot be resolved by the given mesh.

With this discretization of $L^2(D)$, we define P_1 as the mapping from R^N to \hat{D} ,

$$P_1 v = \sum_{i=1}^N v_i \Phi_i(x), \quad \text{for } v = [v_1, v_2, \dots, v_N]^T. \quad (4.6)$$

P_1 is an isometry because we have chosen $\Phi_i(x)$ to be orthonormal.

Then we consider discretizing the range of T_i using a set of orthonormal functions in $L^2(\Omega)$. For example, we can use orthonormal polynomials,

$$\{H_1(\omega), H_2(\omega), \dots, H_M(\omega)\}. \quad (4.7)$$

Then we define the projection operator P_2 that maps from $L^2(\Omega)$ to R^M ,

$$P_2 H(\omega) = (u_1, \dots, u_M)^T, \quad \text{s.t. } \forall j, \quad \left(H(\omega) - \sum_{i=1}^M u_i H_i(\omega) \right) \perp H_j(\omega). \quad (4.8)$$

With the discretization of $L^2(D)$ and $L^2(\omega)$ using P_1 and P_2 , T_i is discretized to

$$\hat{T}_i = P_2 T_i P_1, \quad (4.9)$$

which is a linear map from R^N to R^M thus is a matrix. M and N should be large to make sure the discretization error introduced in (4.4) and (4.7) is small.

We want to approximate the range of \hat{T}_i , which means finding a matrix Q_i with orthonormal columns, such that $\|(I - Q_i Q_i^T) \hat{T}_i\|$ in the operator norm is small. Note that $Q_i Q_i^T$ is the projection operator to the column space of Q_i . Once we get $Q_i = (q_i^{ls})$, which is a $M \times k_i$ matrix, we can construct the corresponding stochastic basis $\xi_i^j(\omega)$ approximating the range of T_i based on the operator P_2 ,

$$\xi_i^j(\omega) = \sum_{l=1}^M H_l(\omega) q_i^{lj}, \quad j = 1, \dots, k_i,$$

where $H_l(\omega)$ are the orthonormal basis functions in $L^2(\Omega)$ (4.7).

We apply the randomized range finding algorithms [19, 24] to \hat{T}_i to construct the Q_i .

4.3 The Randomized Range Finding Algorithms

Given a large matrix $A = \hat{T}_i$ and a tolerance ϵ , we want to construct a matrix Q with orthonormal columns, whose column space can capture the main action of A , namely (in operator norm)

$$\|(I - QQ^T)A\| \leq \epsilon, \quad (4.10)$$

where $I - QQ^T$ is the projection operator to the orthogonal complement of the column space of Q .

The basic idea of the randomized range finding algorithm is using the operator A to act on a random matrix Ω and extracting the main action of A from the image $A\Omega$. The following lemma [19] can be used *a posteriori* to confirm that (4.10) holds (with high probability).

Lemma 4.1. *Let B be a real $m \times n$ matrix. Fix a positive integer r and a real number $\alpha > 0$. Draw an independent family of standard Gaussian vectors*

$$\{\omega^{(i)} : i = 1, 2, \dots, r\}.$$

Then

$$\|B\| \leq \alpha \sqrt{\frac{2}{\pi}} \max_{i=1, \dots, r} \|B\omega^{(i)}\| \quad (4.11)$$

except with probability α^{-r} .

Algorithm 4.1 Adaptive Randomized Range Finder

- 1: Draw standard Gaussian vectors $\omega^{(1)}, \dots, \omega^{(r)}$ of length N .
 - 2: For $i = 1, 2, \dots, r$ compute $y^{(i)} = A\omega^{(i)}$.
 - 3: $j = 0$.
 - 4: $Q^{(0)} = []$, the $m \times 0$ matrix.
 - 5: **while** $\max\{\|y^{(j+1)}\|, \|y^{(j+2)}\|, \dots, \|y^{(j+r)}\|\} > \epsilon/(10\sqrt{2/\pi})$ **do**
 - 6: $j = j + 1$.
 - 7: Overwrite $y^{(j)}$ by $(I - Q^{(j-1)}(Q^{(j-1)})^T)y^{(j)}$.
 - 8: $q^{(j)} = y^{(j)}/\|y^{(j)}\|$.
 - 9: $Q^{(j)} = [Q^{(j-1)} q^{(j)}]$.
 - 10: Draw a standard Gaussian vector w^{j+r} of length n .
 - 11: $y^{(j+r)} = (I - Q^{(j)}(Q^{(j)})^T)Aw^{(j+r)}$.
 - 12: **for** $i = (j+1), (j+2), \dots, (j+r-1)$, **do**
 - 13: Overwrite y^i by $y^{(i)} - q^{(j)}\langle q^{(j)}, y^{(i)} \rangle$.
 - 14: **end for**
 - 15: **end while**
 - 16: $Q = Q^{(j)}$
-

Algorithm 4.1 [19] produces an orthonormal Q such that (4.10) holds with probability at least $1 - \min\{M, N\}10^{-r}$. The Q is obtained by Gram-Schmidt orthogonalization in Step 7, 8, 9 to the image $A\Omega$. Lemma 4.1 is used in Step 5 with $B = (I - QQ^T)A$, $\omega^{(i)} = \omega^{j+i}$, $i = 1, \dots, r$ to check whether the returned Q makes (4.10) hold. The error estimation in this Algorithm is almost free since the computed $(A - Q^TQA)\omega^{j+1}$ can be put back in Q after normalization if (4.11) is not satisfied. Another strategy to extract the main action of A from the image $A\Omega$ is by Singular Value Decomposition as [25]. This strategy together with error estimation using Lemma 4.10 leads to Algorithm 4.2, which produces an orthonormal Q such that (4.10) holds with probability at least $1 - 2\min\{M, N\}10^{-r}$. In Step 9, 10, 11, the first several left singular vectors of $W = A\Omega$ are selected to make (4.1) hold. If all the left singular vectors of W cannot make (4.1) hold in Step 10, W will be enriched in Step 15.

Algorithm 4.2 is more expensive than Algorithm 4.1 because of the SVD process, which might need to be done several times. But based on our numerical tests, Algorithm 4.2 in general returns a relatively smaller Q because of the SVD procedure. We actually implement both Algorithms in the HSFEM framework, and their performances are very similar. For clarity we use Algorithm 4.2 to demonstrate the implementation and efficiency of our method for the rest of this paper. We remark that the estimate (4.11) is pessimistic[19], so in practice a small r in the *a posteriori* estimation is enough.

4.4 Implementation Details of the Sampling Method

We apply Algorithm 4.2 to the operator \hat{T}_i , $i = 1, \dots, n$ to construct the local stochastic basis. Note that T_i is defined as a solution operator to equation (1.1), so we do not have direct access to each entry

Algorithm 4.2 Construct An Orthonormal Matrix Q to Approximate the Range of A

```

1: Draw standard Gaussian vectors  $\omega^{(1)}, \dots, \omega^{(r)}$  of length  $N$ .
2: For  $i = 1, \dots, r$  compute  $y^{(i)} = A\omega^{(i)}$ .  $Y^{(i)} = y^{(i)}$ .
3:  $Q = \emptyset$ .
4: Draw Gaussian matrix  $\Omega$  of size  $N \times K$ .  $W = A\Omega$ .
5: while  $\max\{\|Y^{(1)}\|, \|Y^{(2)}\|, \dots, \|Y^{(r)}\|\} > \epsilon/(10\sqrt{2/\pi})$  do
6:   Do singular value decomposition to  $W$ .  $W = \sum_{i=1}^K \sigma_i U_i V_i^T$  with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_K$ .
7:    $\gamma = 1$ .
8:   while  $\gamma \leq K$  do
9:     For  $i = 1, \dots, r$  compute  $Y^{(i)} = Y^{(i)} - U_\gamma U_\gamma^T Y^{(i)}$ .
10:    if  $\max\{\|Y^{(1)}\|, \|Y^{(2)}\|, \dots, \|Y^{(r)}\|\} < \epsilon/(10\sqrt{2/\pi})$  then
11:      Let  $Q = [U_1, U_2, \dots, U_\gamma]$ . return
12:    end if
13:     $\gamma = \gamma + 1$ .
14:  end while
15: Draw Gaussian  $\hat{\Omega}$  of size  $N \times K$ .  $W = [W, A\hat{\Omega}]$ .
16:  $K = 2K$ .
17: For  $i = 1, \dots, r$ ,  $Y^{(i)} = y^{(i)}$ .
18: end while

```

of \hat{T}_i . However, for any $v = (v_1, \dots, v_N)^T \in R^N$, based on (4.6), (4.9) and (4.1), we can compute $\hat{T}_i v$ by solving equation (1.1) using the forcing function

$$f(x) = \sum_{i=1}^n v_i \Phi_i(x).$$

Denote the solution as $u(x, \omega)$, then we have

$$\hat{T}_i v = P_2(u(x_i, \omega) - E[u(x_i, \omega)]), \quad (4.12)$$

where P_2 is defined in (4.8), and this means we have access to matrix-vector multiplication of \hat{T}_i .

Fortunately, Algorithm 4.2 does not require access to each entry of $A = \hat{T}_i$. It only involves using the operator to act on a random matrix $\hat{\Omega}$, which we have access to based on (4.12), and doing singular value decomposition to the image. We never need to implement the projection operator P_2 in practice. Instead we view the columns of the image $W = A\Omega$ as $L^2(\Omega)$ vectors and do Singular Value Decomposition in Step 6 and Gram-Schmidt orthogonalization in Step 9 according to the $L^2(\Omega)$ inner product.

To avoid doing SVD more than once, we choose the initial K in Step 4 large enough, such that the main action of A can be captured in W in Step 4, i.e., W does not need to be enriched in Step 15. The implementation details of this sampling method are given below:

1. Generate a Gaussian matrix Ω of size $N \times r$, and get the corresponding r forcing functions based on the operator P_1 ,

$$f^p(x) = \sum_{q=1}^N \Omega(q, p) \Phi_q(x), \quad p = 1, \dots, r.$$

Solve Equation (1.1) using forcing $f^p(x)$, $p = 1, \dots, r$, and denote the solutions as $u^p(x, \omega)$. Let

$$Y_i^p(\omega) = u^p(x_i, \omega) - E[u^p(x_i, \omega)].$$

2. Generate a Gaussian matrix Ω of size $N \times K$, and get the corresponding K forcing functions based on the operator P_1 ,

$$f_p(x) = \sum_{q=1}^N \Omega(q, p) \Phi_q(x), \quad p = 1, \dots, K.$$

This means we are sampling each operator \hat{T}_i using the same random matrix Ω . Solve equation (1.1) with forcing $f_p(x)$, $p = 1, \dots, K$, and denote the solutions as $u_p(x, \omega)$. Let

$$y_i^{(p)}(\omega) = u_p(x_i, \omega) - E[u_p(x_i, \omega)], \quad i = 1, \dots, N, \quad p = 1, \dots, K.$$

3. For each node point x_i , compute the $K \times K$ matrix C_i as

$$C_i(p, q) = \int_{\Omega} y_i^{(p)}(\omega) y_i^{(q)}(\omega) dP,$$

which is symmetric and positive semi-definite.

4. For each i , compute the eigen-decomposition of C_i , which is

$$C_i = \sum_j^K \lambda_i^j V_i^j (V_i^j)^T,$$

where V_i^j are $K \times 1$ matrix. C_i can be decomposed efficiently if K is small.

5. Denote the p -th entry of V_i^j as $V_i^j(p)$, then we get the stochastic basis functions for each node x_i .

$$\xi_i^j(\omega) = \sqrt{\frac{1}{\lambda_i^j}} \sum_{l=1}^K V_i^j(p) y_i^{(p)}(\omega), \quad j = 1, \dots, K.$$

6. For each i , find the smallest $\gamma = k_i$ such that for all $p = 1, \dots, r$,

$$\|Y_i^p(\omega) - \sum_{j=1}^{\gamma} \int_{\Omega} Y_i^p(\omega) \xi_i^j(\omega) dP \xi_i^j(\omega)\| \leq \epsilon / (10\sqrt{2/\pi}).$$

Then the local stochastic basis functions $\xi_i^j(\omega)$ at node x_i are

$$\xi_i^j(\omega), \quad j = 1, \dots, k_i.$$

They have mean 0, variance 1, and are orthogonal to each other.

Remark 4.2. The constructed local stochastic basis functions are not exactly the first several left singular vectors of T_i , but to obtain certain accuracy in $L^2(\Omega)$, the number of returned local stochastic basis functions is close to the optimal k_i if the singular values of T_i decay fast [19].

Remark 4.3. Following the same argument as Lemma 2.1, we know that there is high probability that the trial space V_h constructed using the sampling method has the following approximation property,

$$\inf_{v(x, \omega) \in V_h} \|u(x, \omega) - v(x, \omega)\|_{L^2(D \times \Omega)} \leq C\epsilon. \quad (4.13)$$

Note that according to (3.4), the HSFEM searches the best approximation of the solution within the trial space V_h in $L^2(H_0^1(D), \Omega)$, not $L^2(D \times \Omega)$. So the approximation property (4.13) cannot guarantee convergence of the numerical solution in $L^2(D \times \Omega)$. Convergence analysis of the HSFEM framework will be given in our future work. Numerical results in section 6 suggest that the local stochastic basis constructed using the sampling method works well within the HSFEM framework.

5 Numerical Implementation of the HSFEM

In this section, we address several issues concerning the implementation of the HSFEM. We first summarize the outline of the whole algorithm and estimate the main computational cost. Methods to reduce the offline computational cost and an online error estimation and correction procedure are given. Then we discuss two ways of discretization in the stochastic direction. The HSFEM is parallel in nature and can be easily implemented on a parallel machine to attain more computational savings.

5.1 Outline of the Whole Method and Main Computational Cost

The HSFEM involves two stages: the offline stage and the online stage. In the offline stage, we construct the local stochastic basis by sampling the operator using randomly generated forcing functions for a number of times and form the stiffness matrix. In the online stage, we solve equation (1.1) efficiently for multiple forcing functions using the coupling basis constructed from the offline stage.

The **offline stage** involves the following procedures:

- Construct the local stochastic basis functions using Algorithm 4.2, and denote them as

$$\xi_i^j(\omega), \quad i = 1, \dots, n, \quad j = 0, \dots, k_i.$$

The implementation details are given in the previous section.

- Construct the coupling basis $\{\phi_i(x)\xi_i^j(\omega)\}$ and compute the stiffness matrix SM .
 - Let S be the number of coupling basis functions $S = \sum_{i=1}^n k_i + n$, then SM is of size $S \times S$.
 - Let R be the relabeling function that maps each pair $(i, j), i = 1, \dots, n, j = 0, \dots, k_i$ to the global index of the coupling basis function $\phi_i(x)\xi_i^j(\omega)$: $R(i, j) = \sum_{l=1}^{i-1} (k_l + 1) + j + 1$.
 - Compute the stiffness matrix as

$$SM(R(i_1, j_1), R(i_2, j_2)) = \int_{\Omega} \int_D \xi_{i_1}^{j_1}(\omega) \nabla \phi_{i_1}(x)^T a(x, \omega) \xi_{i_2}^{j_2}(\omega) \nabla \phi_{i_2}(x) dx dP. \quad (5.1)$$

Since $\phi_i(x)$ has only local support, the stiffness matrix is sparse.

The offline stage could be quite expensive because in constructing the local stochastic basis, we need to solve equation (1.1) with randomly chosen forcing functions using traditional methods for K times and do singular value decomposition to $\hat{T}_i \hat{\Omega}$, which has K columns. However, if the singular values of operator T_i (4.1) decay fast, a small K is enough for the construction of the local stochastic basis. Our numerical results in section 6 demonstrate that this indeed holds for elliptic operators with several types of stochastic input. For K small, the **main computational cost** in the offline stage comes from sampling the stochastic operator K times. It is of order

$$O(KMn^2), \quad (5.2)$$

where M is the number of sampling points (we use the SC or MC method in our numerical examples in section 6). On each sampling point, we need to solve a deterministic elliptic equation using $\phi_i(x)$. We have assumed the computational cost of solving a sparse $n \times n$ linear system is $O(n^2)$ in (5.2).

In the **online stage** we consider solving the equation with F different forcing functions:

- For each forcing function $f(x)$, construct the load vector b , which is of size $S \times 1$.

$$b(R(i, j)) = \int_{\Omega} \int_D \xi_i^j(\omega) \phi_i(x) f(x) dx dP. \quad (5.3)$$

The local stochastic basis constructed using the sampling method in the previous section has mean 0, so only the $R(i, 0)$ th entries, $i = 1, \dots, n$, of the load vector b are non-zero.

- Solve c_i^j from the linear system

$$SM \times c = b. \quad (5.4)$$

Then the numerical solution is

$$u_h(x, \omega) = \sum_{i=1}^n \sum_{j=1}^{k_i} c_i^j \phi_i(x) \xi_i^j(\omega). \quad (5.5)$$

- Compute the quantities of interest based on the numerical solution $u_h(x, \omega)$.

Since the local stochastic basis functions at each node are orthonormal, we can compute the mean and variance of the solution efficiently without assembling the local stochastic basis functions.

$$E[u(x_i, \omega)] = c_i^0, \quad \sigma^2[u(x_i, \omega)] = \sum_{j=1}^{k_i} (c_i^j)^2.$$

In the online stage, the main computation cost comes from solving the linear system (5.4). If the average number of stochastic basis functions for each node, k , is small, then the linear system (5.4) is small ($kn \times kn$) and sparse, thus can be solved efficiently. The total online computational cost is of order

$$O(Fk^2n^2), \quad (5.6)$$

in which we again assume sparse linear system of size $kn \times kn$ can be solved in $O(k^2n^2)$ operations.

Combining (5.2) and (5.6), we get that the total computational cost of the HSFEM is

$$C_1 K M n^2 + C_2 F k^2 n^2, \quad (5.7)$$

where C_1 and C_2 are two generic constants. On the other hand, if the equation is solved using SC methods for F times, the computational cost is

$$C_1 F M n^2. \quad (5.8)$$

The computational cost of SC method and the HSFEM in the multi-query setting is illustrated in Figure 1. The T in the vertical axis is the offline computational cost of the HSFEM, $T \approx C_1 K M n^2$. The F^* in the horizontal axis is the critical number of queries when the cost of SC and the HSFEM are equal, $F^* \approx C_1 K M n^2 / (C_1 M n^2 - C_2 k^2 n^2)$. The slope of the blue line $k^2 n^2$ is much less than that of the red line $M n^2$ when the elliptic operator enjoys the Operator-Sparsity, which means k is small.

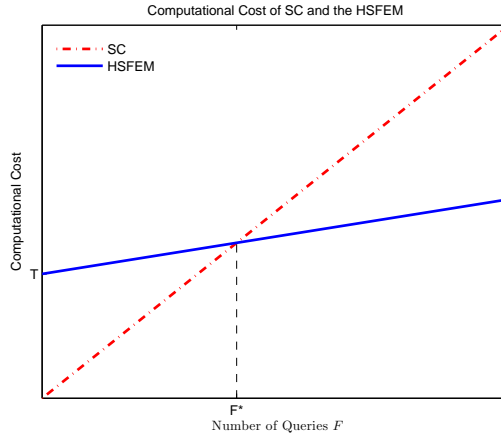


Figure 1: Computational cost of SC methods and the HSFEM in the multi-query setting.

Based on the computational cost (5.7) and (5.8), we can see that:

- If the stochastic operator enjoys the Operator-Sparsity, which means k is small, the online computation can be very efficient since the linear system we solve is sparse and small.
- Our method can achieve computational savings only if $F > F^*$.

If the equation (1.1) only needs to be solved once, then our method cannot bring in any computational savings because of the offline computation. However, if we need to solve it multiple times with different forcing functions, then our method can be very efficient.

5.2 Methods to Reduce the Offline Computational Cost and An Online Error Estimation and Correction Procedure

The offline computational cost is expensive since it requires solving equation (1.1) K times with randomly chosen forcing functions using a traditional method. We discuss methods to reduce the offline computational cost and an online error estimation and correction procedure in this subsection.

The following strategies can be considered to reduce the offline computational cost:

- Construct the local stochastic basis on a coarser mesh.

We can use a coarser mesh in the offline stage to reduce the n in (5.2). To be specific, let $x^i, i = 1, \dots, n_c$ be a coarse mesh and $\phi^i(x), i = 1, \dots, n_c$ be the corresponding coarse mesh piecewise linear basis. We use $\phi^i(x)$ to discretize the equation in the offline stage and the sampling solutions are represented by $\phi^i(x)$. We restrict them to the fine mesh nodes $x_i, i = 1, \dots, n$, and then follow the implementation details in section 4.4 to construct the local stochastic basis functions associated with x_i . Similar strategy has been employed to reduce computational cost in [9, 12].

- Choose a relatively large ϵ in the randomized range finding algorithm.

In certain cases, the singular values of operator T_i do not decay very fast, and a large number of local stochastic basis functions are required to obtain accuracy ϵ , thus the required K in the offline stage is large. To reduce the offline computational cost, we can choose a relative larger ϵ , thus reduce the K in the computational cost (5.2). Note that the constructed local stochastic basis using the larger ϵ is not as accurate, and the online numerical solutions may have large error.

- Choose a small K in the randomized range finding algorithm.

It will be shown in section 6 that the solution space to (1.1) has spatially heterogeneous stochastic structure, and the required number of local stochastic basis functions k_i are different for different nodes x_i . If for certain (small) region of the domain, the required number of local stochastic basis functions k_i is very large, then a large K is required in the offline stage. However, for other regions of the domain, a smaller K may be enough in Algorithm 4.2. To reduce the computational cost, we can choose a relatively small K . And in the Step 8-14 of Algorithm 4.2, if all the left singular vectors of $A\Omega$ cannot make (4.1) hold, we simply return the Algorithm with $Q = [U_1, \dots, U_K]$.

The strategies mentioned above can reduce the offline computational cost to some degree, but the resulting local stochastic basis is not as accurate, and the corresponding online numerical solution may have large error. Here we introduce a procedure of online error estimation and correction [9] through Monte Carlo sampling. We want to emphasize that this *a posteriori* error estimation and correction procedure can be incorporated into the HSFEM framework to get more faithful numerical results even if the above strategies to reduce offline computational cost are not taken.

In many practical UQ problems, we care about the statistical quantity of the solution $u(x, \omega)$, which we denote by $E[g(u(x, \omega))]$, where g is a functional defined on realizations of the stochastic solution. Denote $\omega^i, i = 1, \dots, N_{MC}$ as N_{MC} Monte Carlo sampling points, then $E[g(u(x, \omega))]$ can be approximated by

$$I_{N_{MC}}[g(u)] = \frac{1}{N_{MC}} \sum_{n=1}^{N_{MC}} g(u(x, \omega^n)). \quad (5.9)$$

We denote the error of the Monte Carlo estimator (5.9) as

$$\epsilon_{g(u)}(N_{MC}) = E[g(u)] - I_{N_{MC}}[g(u)].$$

Then $\epsilon_g(N_{MC})$ is a random variable, and according to the central limit theorem, the root-mean-square error of the Monte Carlo estimator (5.9) decays with N_{MC} like

$$\sqrt{E[\epsilon_{g(u)}^2(N_{MC})]} = \frac{\sigma[g(u)]}{\sqrt{N_{MC}}},$$

where $\sigma[g(u)]$ is the standard deviation (SD) of $g(u)$. Note that the decay rate of the error $\frac{1}{\sqrt{N_{MC}}}$ is low, and this limits the application of MC methods. One way to accelerate the convergence is reducing the variance of $g(u)$. Using the HSFEM numerical solution $u_h(x, \omega)$, we can divide $E[g(u)]$ as

$$E[g(u(x, \omega))] = E[g(u_h(x, \omega))] + E[g(u(x, \omega)) - g(u_h(x, \omega))] \quad (5.10)$$

$$\approx E[g(u_h)] + \sum_{k=1}^{N_{MC}} \frac{1}{N_{MC}} (g(u(x, \omega^k)) - g(u_h(x, \omega^k))). \quad (5.11)$$

The first part in (5.11) can be efficiently computed using the HSFEM in the online stage. As to the second part, which is a Monte Carlo estimator, we have that its root-mean-square error decays as

$$\sqrt{E[\epsilon_{g(u)-g(u_h)}^2(N_{MC})]} = \frac{\sigma[g(u) - g(u_h)]}{\sqrt{N_{MC}}},$$

where $\sigma[g(u) - g(u_h)]$ is the SD of $g(u) - g(u_h)$.

We make a **key assumption** here that the numerical solution u_h using the HSFEM, even though may be not accurate, can still capture the main stochastic part of $u(x, \omega)$. Then $g(u) - g(u_h)$ has significantly smaller variance than $g(u)$, and much fewer MC samples in approximating $E[g(u) - g(u_h)]$ in (5.11) than in (5.9) is enough to obtain high accuracy. From this point of view, the HSFEM can be viewed as a tool for **variance reduction**. We denote $\tau(\omega) = g(u) - g(u_h)$, which has relatively small variance, and set a threshold ϵ . Then the error correction procedure includes:

1. Generate N_{MC} MC samples, $\omega^k, k = 1, \dots, N_{MC}$.

2. Denote $\bar{\tau}$, and $\tilde{\tau}$ as

$$\bar{\tau} = \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \tau(\omega^k), \quad \tilde{\tau} = \sqrt{\frac{1}{N_{MC}(N_{MC}-1)} \sum_{k=1}^{N_{MC}} (\tau(\omega^k) - \bar{\tau})^2}. \quad (5.12)$$

Then an approximate 95%-level confidence interval of $E[\tau(\omega)]$ is given by $[\bar{\tau} - 2\tilde{\tau}, \bar{\tau} + 2\tilde{\tau}]$.

3. Consider the following three cases:

- $(|\bar{\tau}| + 2|\tilde{\tau}|)/|E[g(u_h)]| \leq \varepsilon$, which means the numerical result using the HSFEM is accurate. Then we use $E[g(u_h)]$ as a faithful approximation of $E[g(u)]$.
- If the first case does not hold, but $\tilde{\tau} < \varepsilon|E[g(u_h)] + \bar{\tau}|$, which means the error in the estimate of $E[\tau]$ is negligible. Then we use $E[g(u_h)] + \bar{\tau}$ as a faithful approximation of $E[g(u)]$.
- If neither of the above two cases holds, which means the number of MC samples N_{MC} is not large enough, then we double the sampling number N_{MC} and go back to step 1.

5.3 Discretization in the Stochastic Direction

In the spatial direction, the domain D is discretized using a mesh with nodes $x_i, i = 1, \dots, n$, and the solution is represented using the standard piecewise linear basis, $\phi_i(x), i = 1, \dots, n$. While in the stochastic direction, there are basically two ways of discretization:

- Represent the local stochastic basis and numerical solution using a set of orthonormal polynomials. If in the offline stage we use PC methods to solve equation (1.1), or if we use SC methods to solve the equation and interpolate the solution using orthonormal polynomials, then the local stochastic basis and numerical solutions will be represented in this way.
- Discretize (Ω, P) using sampling points $\theta^1, \theta^2, \dots, \theta^M$ with weights w^1, w^2, \dots, w^M . If in the offline stage we use SC methods or MC methods to solve equation (1.1), then the numerical solutions and local stochastic basis functions are represented in this way. In MC methods $w^i = \frac{1}{M}$, while in SC methods θ^i and w^i are determined by the underlying probability measure.

In our numerical examples, we use the second way to discretize the problem in the stochastic direction, i.e., in the offline stage we solve the equation on sampling points $\theta^i, i = 1, \dots, M$ and the local stochastic basis functions are stored as length- M vectors. In computing the stiffness matrix (5.1) and load vector (5.3), we use the same set of sampling points θ^i and weights w^i for numerical integration.

The Smolyak sparse grid quadrature [28, 31] can alleviate the *curse of dimensionality* to some degree when the dimension of the stochastic input is not very high and the solution has high regularity in the stochastic direction. In our numerical examples we use the Smolyak sparse grid to generate θ^i and w^i when the dimension of the stochastic input is not very high. Otherwise we use Monte Carlo method to generate θ^i and set the weights $w^i = \frac{1}{M}$, for $i = 1, \dots, M$.

Under this discretization, our final numerical solution $u_h(x, \omega)$ (5.5) is also defined on the sampling points θ^i . If θ and w are chosen to be the sparse grid collocation points, we can recover $u(x, \omega)$ for any $\omega \in \Omega$ using polynomial interpolation.[28] When MC sampling points are used, we can **only** compute the statistical quantities of the solutions based on their values on these sampling points.

5.4 Parallelization of the HSFEM

The offline computation cost of the HSFEM can be expensive if a large K is chosen, but the HSFEM is parallel in nature and can be easily implemented on a parallel machine. To be specific:

- The offline stage involves solving equation (1.1) for a number of times using randomly generated forcing functions. These forcing functions are independent, thus this process can be parallelized.
- In the construction of local stochastic basis, we need to do singular value decomposition on each node point. And these decompositions are independent, thus can be parallelized.
- In the online stage, after solving equation (5.4) and getting the coefficients \mathcal{C}_i^j , we need to assemble the coupling basis functions $\phi_i(x)\xi_i^j(\omega)$ to recover the numerical solution $u_h(x, \omega)$ (5.5). The assembling is independent on each node point thus can be parallelized.

6 Numerical Results

In this section, we present numerical results to demonstrate that:

- The solution space to stochastic elliptic equation has spatially heterogeneous stochastic structure, and this can be recognized by the HSFEM in the offline stage.
- Several elliptic operators with high dimensional stochastic input enjoys the Operator-Sparsity.
- The local stochastic basis constructed using the sampling method works well within the HSFEM framework. And the HSFEM can take advantage of the sparsity of the operator to attain significant computational savings while maintaining high accuracy in the online stage.

To quantify the accuracy of our numerical solution, we need to compare it with the exact solution. However in most cases it is impossible to construct the exact solution analytically, so we instead choose a suitable numerical solution as a reference. In our numerical implementation, we discretize the domain D using a piecewise linear basis $\phi_i(x)$, and discretize (Ω, P) using sampling points θ^i with weights w^i . Then the error in our numerical solution $u_h(x, \omega)$ can be divided into two parts as

$$u_h(x, \omega) - u(x, \omega) = [u_d(x, \omega) - u(x, \omega)] + [u_h(x, \omega) - u_d(x, \omega)], \quad (6.1)$$

where $u_d(x, \omega)$ is the numerical solution to equation (1.1) using the SC (or MC) method. $u_d(x, \omega)$ is obtained by solving equation (1.1) on the discrete sampling points $\theta^i, i = 1, \dots, M$, using the spatial basis $\phi_i(x), i = 1, \dots, n$. The first part of the error in (6.1) is the **discretization error**, which comes from the discretization of D and Ω . The second part of (6.1) comes from our selection of finite local stochastic basis functions from $L^2(\Omega)$. For the purpose of evaluating the performance of the HSFEM, we ignore the discretization error and only consider the second part of the error, namely, we use $u_d(x, \omega)$ as the reference solution. In the rest of this section we denote $u_d(x, \omega)$ as $u(x, \omega)$ for simplicity.

In each of our numerical examples in this section we compute the average number of the local stochastic basis functions constructed in the offline stage, which is

$$k = \sum_{i=1}^n \frac{k_i}{n}.$$

If k is small, and the numerical error using the HSFEM is small, then a small set of coupling basis $\{\phi_i(x)\xi_i^j(\omega)\}$ can approximate the solution space well. Thus the stochastic elliptic operator that we consider enjoys the Operator-Sparsity. Besides, the linear system we solve in the online stage is of size $n(k+1) \times n(k+1)$, so smaller k also implies better efficiency of the HSFEM.

We also compute the error in Expectation (mean) and Standard Deviation (SD) of the solution, which are the two primary quantities of interest in Uncertainty Quantification:

$$\begin{aligned} E[u] - E[u_h] &= \sum_{i=1}^M w^i (u(x, \theta^i) - u_h(x, \theta^i)). \\ \sigma[u] - \sigma[u_h] &= \left(\sum_{i=1}^M w^i (u(x, \theta^i) - E[u(x, \omega)])^2 \right)^{1/2} - \left(\sum_{i=1}^M w^i (u_h(x, \theta^i) - E[u_h(x, \omega)])^2 \right)^{1/2}. \end{aligned}$$

We compute the relative $L^2(D \times \Omega)$ error in the stochastic part of the solution,

$$E_{HSFEM} = \frac{\left(\sum_{i=1}^M w^i \|u(x, \theta^i) - u_h(x, \theta^i)\|_{L^2(D)}^2 \right)^{\frac{1}{2}}}{\left(\sum_{i=1}^M w^i \|u(x, \theta^i) - \bar{u}(x)\|_{L^2(D)}^2 \right)^{\frac{1}{2}}}. \quad (6.2)$$

Note that in the denominator is the $L^2(D \times \Omega)$ norm of $u(x, \omega) - \bar{u}(x)$, not $u(x, \omega)$, so E_{HSFEM} measures the capacity of the HSFEM in capturing the stochastic part of the solution.

6.1 Comparison with the KL Expansion

The key difference of the HSFEM framework from traditional methods is that different stochastic basis functions are used in different regions of the domain to approximate the stochastic part of the solution.

By doing so we allow for spatially heterogeneous stochastic structure of the solution space and expect to reduce the total degrees of freedom and computational cost in the online stage.

To demonstrate the advantage of approximating the stochastic behavior of the solution locally, we compare the accuracy of the HSFEM numerical solution with the truncated KL expansion of the exact solution, which, as we introduced in section 2.1, is the **optimal approximation** to the stochastic part of the solution in $L^2(D \times \Omega)$ when global stochastic basis functions are used.

Recall that the numerical solution using the HSFEM $u_h(x, \omega)$ is represented as

$$u_h(x, \omega) = \sum_{i=1}^n \sum_{j=0}^{k_i} c_i^j \phi_i(x) \xi_i^j(\omega),$$

in which a total number of

$$S = \sum_{i=1}^n (1 + k_i)$$

basis functions are used. On the other hand, consider the \tilde{k} -term KL expansion of $u(x, \omega)$,

$$u_{\tilde{k}}(x, \omega) = \bar{u}(x) + \sum_{j=1}^{\tilde{k}} \sqrt{\lambda_j} \psi_j(x) \xi_j(\omega) = \sum_{i=1}^n \bar{u}(x_i) \phi_i(x) + \sum_{j=1}^{\tilde{k}} \sum_{i=1}^n \sqrt{\lambda_j} \psi_j(x_i) \phi_i(x) \xi_j(\omega), \quad (6.3)$$

in which a total number of

$$n + \tilde{k}n$$

basis functions are used. To make a fair comparison between the HSFEM numerical solution and the truncated KL expansion, we should keep the degrees of freedom the same, namely,

$$S = n + \tilde{k}n,$$

which means we choose $\tilde{k} = k = \frac{\sum_{i=1}^n k_i}{n}$ in the truncated KL expansion (6.3) to make the comparison.

Note that in both $u_h(x, \omega)$ and $u_k(x, \omega)$, k (on average) stochastic basis functions are used to approximate the stochastic part of the solution on each node x_i . The difference is that the local stochastic basis functions $u_h(x, \omega)$ uses are different on each node point, while the stochastic basis functions in $u_k(x, \omega)$ are the same on the whole domain. We define the relative KL truncation error as

$$E_{KL} = \frac{\|u(x, \omega) - u_k(x, \omega)\|_{L^2(D \times \Omega)}}{\|u(x, \omega) - \bar{u}(x, \omega)\|_{L^2(D \times \Omega)}}. \quad (6.4)$$

We compare E_{KL} with the numerical error E_{HSFEM} (6.2) for each numerical example in this section. We emphasize that this is an unfair comparison since the stochastic basis in $u_k(x, \omega)$ is adapted to the solution with a specific forcing and does not necessarily approximate another solution well, while the stochastic basis in $u_h(x, \omega)$ is constructed to approximate the whole solution space for all $f(x) \in L^2(D)$.

So if for some forcing function $f(x)$, $E_{HSFEM} \leq E_{KL}$ (or they are of the same order considering the unfairness of this comparison), then it reflects the heterogeneous stochastic structure of the solution, and the advantage of approximating the stochastic behavior of the solution locally.

6.2 A Thoroughly Studied 1D Model Problem

In this subsection we consider the following 1D elliptic SPDE:

$$\begin{cases} -\frac{\partial}{\partial x}(a(x, \omega) \frac{\partial}{\partial x} u(x, \omega)) = f(x), & x \in D = (0, 1), \quad \omega \in \Omega, \\ u(0, \omega) = 0, \quad u(1, \omega) = 0. \end{cases} \quad (6.5)$$

We assume the KL expansion of $\log a(x, \omega)$ is given by

$$\log a(x, \omega) = \sum_{i=1}^m \cos(2\pi i x) \omega_i, \quad (6.6)$$

where ω_i , $i = 1, \dots, m$, are independent random variables,

$$\omega_i \sim \mathcal{U}(-1/2, 1/2).$$

Note that in the KL expansion of $\log a(x, \omega)$, the first m singular values are equal, which means the m random variables ω_i contribute equally to $\log a(x, \omega)$ in the $L^2(D \times \Omega)$ sense and none of them is negligible. So this problem has genuine high stochastic dimension for large m .

6.2.1 The Linear Compact Operator T_i

Recall that T_i (4.1) is a compact linear operator mapping from the forcing $f(x) \in L^2(D)$ to the stochastic part of the solution at x_i , $u(x_i, \omega) - \bar{u}(x_i) \in L^2(\Omega)$. T_i can be discretized to a matrix \hat{T}_i of size $M \times N$ through (4.9). Since T_i is defined as a solution operator, we do not have direct access to each entry of \hat{T}_i , but we have access to matrix-vector multiplication of \hat{T}_i based on (4.12). In this subsection, we use \hat{T}_i to act on the standard basis $e_k, k = 1, \dots, N$, of R^N , by doing which we explicitly get all the columns of \hat{T}_i . We use this explicit T_i to investigate the 1D stochastic elliptic operator.

In the spatial direction we discretize the domain $[0, 1]$ using uniform mesh of size $h = 1/256$. In the stochastic direction, we discretize (Ω, P) using (θ^i, w^i) , which are the 4th order Smolyak sparse grid collocation points. The orthonormal basis functions to discretize $L^2(D)$ (4.3) are chosen to be

$$\{1, \dots, 2 \sin(2\pi l x), 2 \cos(2\pi l x)\} \quad (6.7)$$

with $l = 128$, which contains all the Fourier modes that can be resolved by the given mesh.

We choose $m = 20$ in (6.6), and explicitly compute the T_i with $x_i = 1/2$. The decay of the singular values of T_i , σ_i^k , is plotted in Figure 2. From this figure, we can see that even though the stochastic input ω has high dimension, σ_i^k still decays exponentially fast. The fast-decay of the singular values of T_i reveals the Operator-Sparsity of this 1D stochastic operator. It also justifies the sampling method to construct the local stochastic basis, since the performance of the randomized range finding algorithm depends on the decay rate of the singular values of the operator.[19]

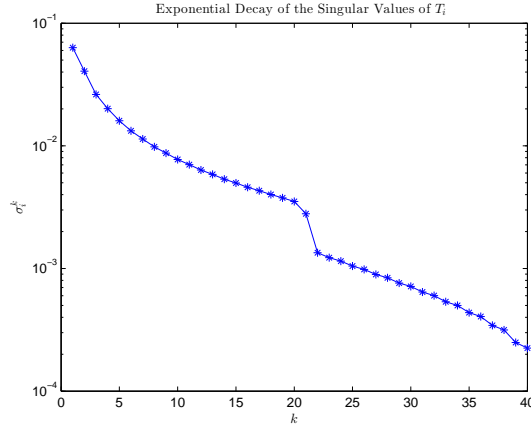


Figure 2: Exponential decay of the singular values of T_i .

The number of required terms to make the truncation error of T_i less than ϵ is

$$k_i = \inf\{k : \sigma_i^k \leq \epsilon\}.$$

We choose m ranging from 10 to 30 in (6.6), and for each m , we compute the number of required terms k_i (with $x_i = 1/2$) to make the truncation error of T_i less than $\epsilon = 2 \times 10^{-3}$. The dependence of k_i on m is plotted in Figure 3a. From this figure we can see that k_i seems to grow unboundedly (about linearly) with the stochastic dimension. This does not contradict our analysis in section 2.3, since as m increases, the ellipticity of the stochastic operator deteriorates, i.e., λ_{\min} decreases and λ_{\max} increases. To confirm the analysis in section 2.3, we consider a normalized model with the coefficient $\tilde{a}(x, \omega)$ given by

$$\log \tilde{a}(x, \omega) = \sum_{i=1}^m \frac{20}{m} \cos(2\pi i x) \omega_i,$$

where $\omega_i, i = 1, \dots, m$ are independent and $\omega_i \sim \mathcal{U}(-1/2, 1/2)$. We can see that for all $m \geq 1$,

$$\tilde{a}(x, \omega) \in [e^{-10}, e^{10}].$$

For this normalized model, we use the same discretization as the unnormalized model and compute the number of required terms k_i to make the truncation error of T_i less than $\epsilon = 2 \times 10^{-3}$. The dependence of k_i on m is plotted in Figure 3b. From this figure, we can see that k_i seems to remain bounded as the stochastic dimension m increases, which agrees our analysis in section 2.3.

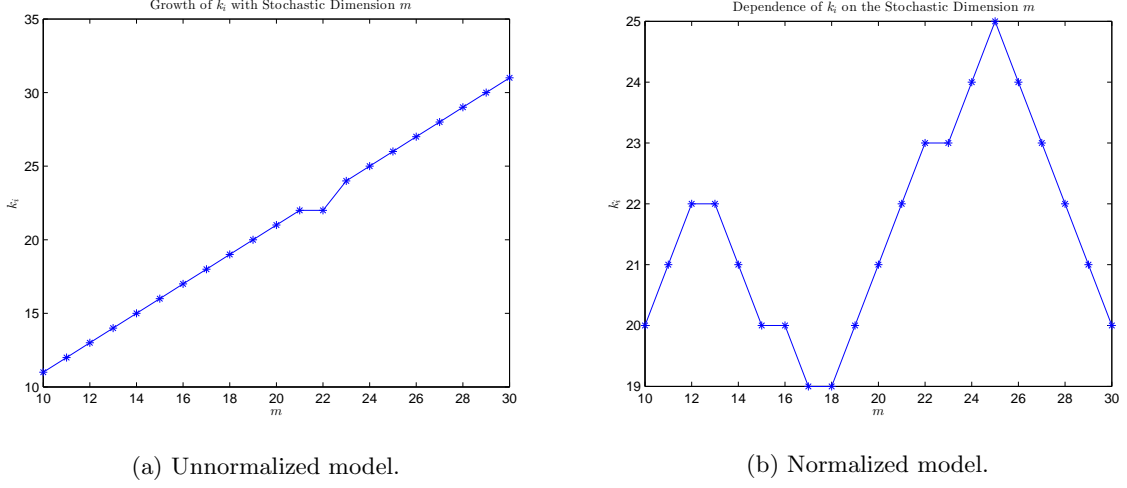


Figure 3: Dependence of k_i on m .

6.2.2 Performance of the Heterogeneous Stochastic Finite Element Method

In this section we apply the HSFEM to solve equation (6.5) with $m = 20$ in (6.6). We use the same discretization in the spatial and stochastic directions as the previous subsection. In the offline stage we discretize $L^2(D)$ as in (6.7) and the ϵ , r and K in Algorithm 4.2 are chosen to be

$$\epsilon = 10\sqrt{2/\pi} \times 10^{-3}, \quad K = 50, \quad r = 5.$$

The average number of local stochastic basis functions constructed in the offline stage is

$$k = \frac{1}{n} \sum_{i=1}^n k_i \approx 26.2.$$

For this $m = 20$ dimensional problem, $k = 26.2$ is quite small compared with the *curse of dimensionality* suffered from by the PC and SC methods. The distribution of k_i is plotted in Figure 4.

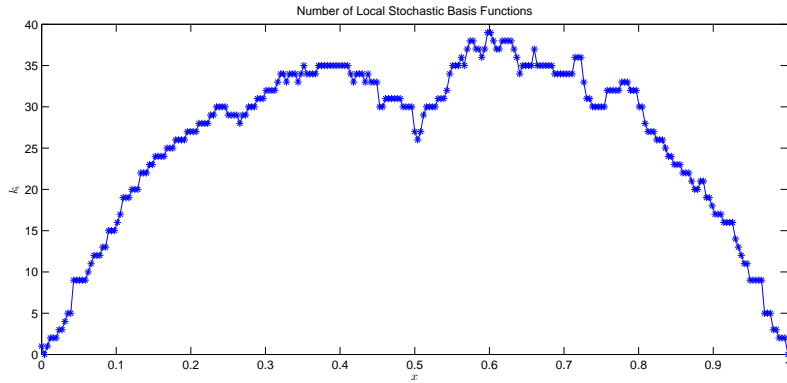


Figure 4: Distribution of local stochastic basis constructed offline.

From this figure we can see that more local stochastic basis functions are constructed in the interior of the domain than near the boundary. This is because we have chosen deterministic (actually homogeneous) boundary condition in equation (6.5), and the solution has less randomness near the boundary.

In the online stage we solve equation (6.5) with forcing function given by

$$f(x) = 1 - x + x^2 - x^3. \quad (6.8)$$

The expectation and standard deviation of the solution, and the numerical errors in these two quantities are plotted in Figure 5. We can see that our method attains high accuracy in both mean and SD

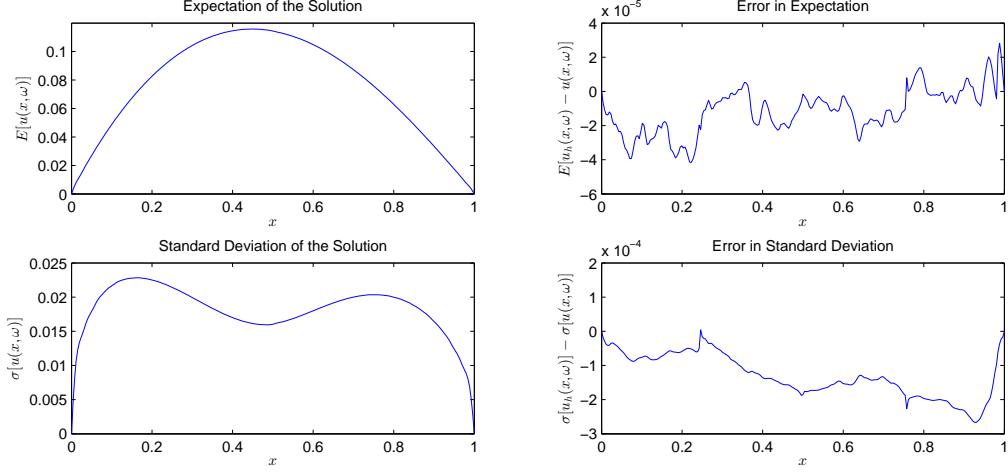


Figure 5: Numerical error in Expectation and Standard Deviation of the solution.

k	Relative $L^2(D \times \Omega)$ Error E_{HSFEM}	Relative KL truncation Error E_{KL}
26.2	1.83×10^{-2}	1.81×10^{-2}

Table 1: Relative $L^2(D \times \Omega)$ error E_{HSFEM} and relative KL truncation error E_{KL} .

of the solution, which reflects that the local stochastic basis constructed using the sampling method in section 4 works well within the HSFEM framework.

The relative $L^2(D \times \Omega)$ error E_{HSFEM} (6.2) and relative KL truncation error E_{KL} (6.4) are listed in Table 1. $E_{HSFEM} = 1.83 \times 10^{-2}$ is small, which means the HSFEM attains high accuracy in capturing the stochastic part of the solution. For this problem, the KL truncation error E_{KL} is of the same order as E_{HSFEM} . As we have argued in section 6.1, this reflects the spatially heterogeneous stochastic structure of the solution and the advantage of approximating the stochastic part of the solution locally.

6.2.3 Convergence rate of the HSFEM

The parameter ϵ in Algorithm 4.2 affects the number of the returned local stochastic basis functions and the accuracy of the online numerical solution. To study how does ϵ affect the numerical error in the online stage, we consider solving the same equation as the previous subsection using different ϵ in the offline stage. Since the HSFEM is randomized in nature, the k and $L^2(D \times \Omega)$ Error are both random. To reduce the fluctuation, for each ϵ , we construct the local stochastic basis and solve the equation for 10 times. The averaged k and $L^2(D \times \Omega)$ Error of the 10 times for different ϵ are plotted in Figure 6.

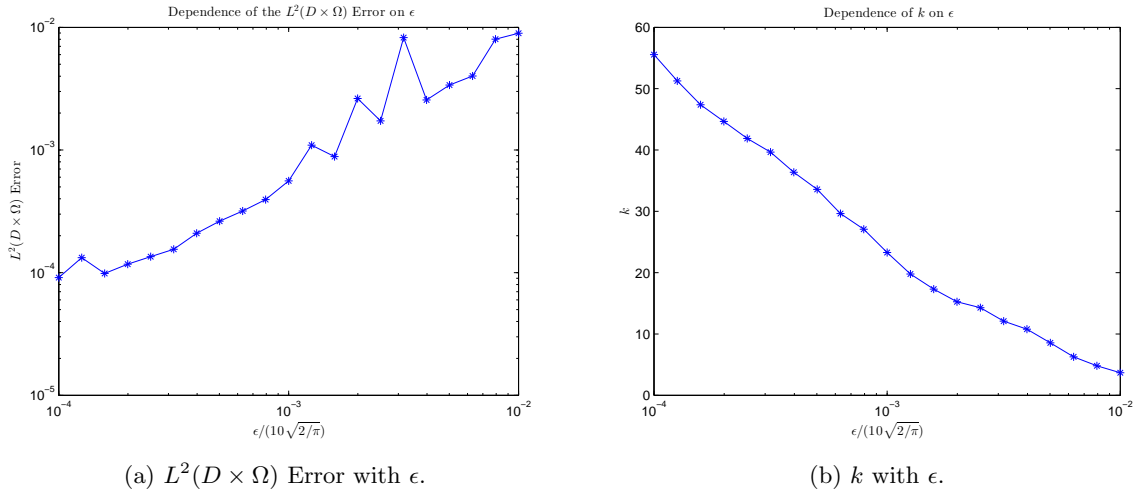


Figure 6: Dependence of $L^2(D \times \Omega)$ Error and k on the parameter ϵ .

From Figure 6a, we can see that the $L^2(D \times \Omega)$ error, which we denote by E , grows about linearly with ϵ . By doing linear regression $\log E = \log C + \alpha \log \epsilon$, we obtain that

$$E \approx C \times \epsilon^{1.08}, \quad C \approx 6.8 \times 10^{-2}.$$

The linear growth of the $L^2(D \times \Omega)$ error with ϵ agrees with the approximation property (4.13).

From Figure 6b, we can see that the returned k grows about linearly with $-\log \epsilon$ as (2.11), which is much milder than (2.9) because the singular values of T_i actually decay exponentially fast.

6.2.4 The Online Error Estimation and correction Procedure

In this subsection, we illustrate the implementation of the strategies to reduce the offline computational cost and the online error estimation and correction procedure introduced in section 5.2. We consider the stochastic elliptic operator given by (6.6) with $m = 30$. For this problem with very high dimensional stochastic input, a very large number of collation points are required to obtain accuracy even if the Smolyak sparse grid quadratures are employed. So we instead use Monte Carlo samplings to discretize the probability space (Ω, P) . We generate $M = 4 \times 10^4$ samples of ω according to its distribution, and store them as $\theta^1, \dots, \theta^M$. We set the weights $w^i = \frac{1}{M}$, then (Ω, P) is discretized to (θ^i, w^i) . With this discretization in the stochastic direction, the local stochastic basis functions are stored as length- M vectors, and the online numerical solutions are also defined on these sampling points. We can compute the statistical quantities of the numerical solutions based on their values on the sampling points and the weights w^i , but cannot recover the whole solution using polynomial interpolation as the previous cases where the stochastic collocation points are used to discretize (Ω, P) .

In the offline stage, we discretize the domain $D = [0, 1]$ using a uniform coarse mesh of size $h = 1/128$ to construct the local stochastic basis. In Algorithm 4.2, we choose

$$\epsilon/10\sqrt{2/\pi} = 3 \times 10^{-3}, \quad K = 35, \quad r = 5.$$

We first solve equation (6.6) with randomly chosen forcing functions using the coarse mesh spatial basis, and then restrict the sampling solutions to the nodes of a uniform fine mesh of size $h = 1/256$ to construct the local stochastic basis using Algorithm 4.2.

The average number of local stochastic basis functions constructed in the offline stage is

$$k = \frac{\sum_{i=1}^n k_i}{n} = 24.8.$$

This k is small, which means in the online stage the linear system that we need to solve is small. The distribution of k_i is plotted in Figure 7. We can see that for about half of the node points, we have $k_i = K$. This is because in Step 7-14 of Algorithm 4.2, all the left singular vectors of $A\Omega$ cannot make the condition in Step 10 hold. And to control the offline computation cost, we did not choose a larger K as in Step 15-18, but simply return the algorithm with $Q = [U_1, \dots, U_K]$.

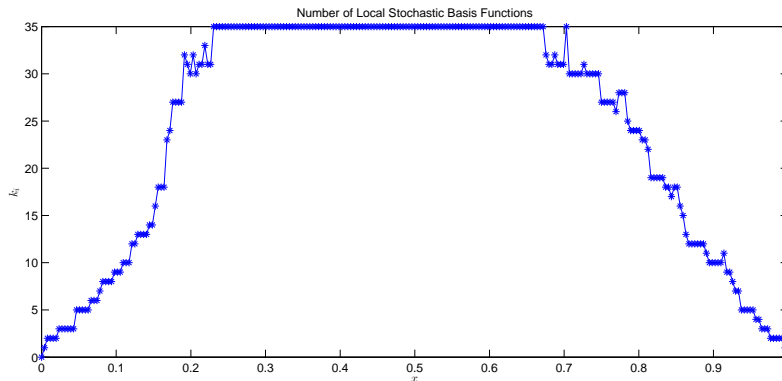


Figure 7: Distribution of local stochastic basis.

The mean and SD of the solution and the numerical errors in these two quantities are plotted in Figure 8. We can see that there are relatively larger errors in these two quantities, and this is because the strategies introduced in section 5.2 are taken to reduce the offline computational cost and the constructed

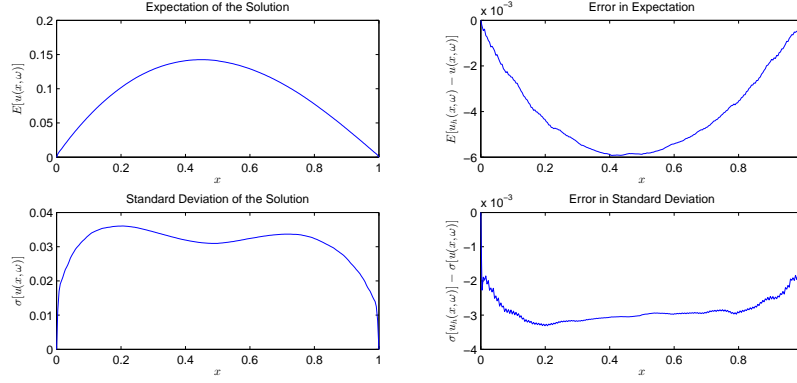


Figure 8: Numerical error in Expectation and Standard Deviation of the solution.

local stochastic basis is not as accurate. For this case, the online error estimation and correction procedure is needed to obtain faithful numerical results.

To illustrate the online error estimation and correction procedure, we consider the second order moment of the solution at $x = 1/2$, i.e., we consider

$$g(u) = u(1/2, \omega)^2.$$

Then based on the online numerical solution using the HSFEM, we have

$$E[g(u_h)] = 1.899 \times 10^{-2}. \quad (6.9)$$

And for the exact solution $u(x, \omega)$, which is the MC solution using the sampling points θ^i since we do not consider discretization error here, we have

$$E[g(u)] = 2.078 \times 10^{-2}. \quad (6.10)$$

We can see that the error in $E[g(u_h)]$ is non-negligible. In practice, we will not compute the exact solution $u(x, \omega)$ because of the expensive computational cost, but instead apply the online error correction procedure given in section 5.2 to correct $E[g(u_h)]$ and get a better approximation of $E[g(u)]$.

Since the HSFEM online numerical solution $u_h(x, \omega)$ is only defined on the discrete sampling points θ^i , we cannot apply MC method to the original equation (6.6) to correct error as in (5.11). Instead, we apply the error estimation and correction procedure to the discretized problem with Ω replaced by $\{\theta^1, \dots, \theta^M\}$ and P replaced by w^1, \dots, w^M . We choose $N_{MC} = 100$ in (5.11), and generate N_{MC} independent random integers i uniformly distributed from 1 to M (because w^i are all equal). We denote them as $i_{n_{mc}}, n_{mc} = 1, \dots, N_{MC}$ and then solve equation (6.6) on these sampling point $\theta^{i_{n_{mc}}}$. The average and variance of $\tau(\omega)$ (5.12) are approximately

$$E[g(u_h)] + \bar{\tau} = 2.062 \times 10^{-2}, \quad \bar{\tau} = 1.025 \times 10^{-4}.$$

Thus we get an approximate 95% level confidence interval for $E[g(u)]$,

$$[2.042 \times 10^{-2}, 2.082 \times 10^{-2}]. \quad (6.11)$$

If we instead apply MC methods directly to the discretized equation (6.6) using the same $N_{MC} = 100$ in (5.9), we get

$$E[g(u)] \approx 2.033 \times 10^{-2}, \quad \sigma[g(u)] = 9.210 \times 10^{-4}.$$

and an approximate 95% level confidence interval for $E[g(u)]$,

$$[1.805 \times 10^{-2}, 2.261 \times 10^{-2}]. \quad (6.12)$$

We can see that the approximation interval (6.11) is much more accurate than (6.12), and this is because the HSFEM solution $u_h(x, \omega)$ can capture the main stochastic part of the solution, and $g(u) - g(u_h)$ has much smaller variance than $g(u)$. Using the discretized exact solution, we can compute that

$$\sigma[g(u)] = 1.2361 \times 10^{-2}, \quad \sigma[g(u) - g(u_h)] = 2.3131 \times 10^{-3}.$$

This example demonstrates the effectiveness of the HSFEM as a tool for variance reduction. We emphasize that, in practice, this online error estimation and correction procedure can be used to get more faithful numerical results even if the strategies to reduce offline cost are not taken.

6.3 A 2D Example With Gaussian Random Variables

In this section, we consider the following 2D SPDE

$$\begin{cases} -\operatorname{div}(a(x, y, \omega) \nabla u(x, y, \omega)) = f(x, y), & (x, y) \in D = (0, 1)^2, \quad \omega \in \Omega, \\ u(x, \omega)|_{\partial D} = 0. \end{cases} \quad (6.13)$$

The coefficient $a(x, \omega)$ is given by

$$\log(a(x, y, \omega)) = 1 + \frac{1}{4} \sum_{k=1}^{12} \omega_k (\sin(k\pi x) + \cos((13-k)\pi y)),$$

where $\omega_i, i = 1, \dots, 12$ are independent Gaussian random variables, $\omega_i \sim \mathcal{N}(0, 1)$. For this coefficient $a(x, \omega)$, the uniform ellipticity condition (1.2) is actually violated. But in our implementation, we discretize Ω using (finite) sampling points, and the discretized problem is still elliptic and well-posed.

None of the 12 random variables is negligible in the coefficient thus this stochastic operator has genuine high stochastic dimension. In the spatial direction, the domain D is discretized using a standard right triangular mesh of size $h = 1/64$, resulting in 2×64^2 elements. In the stochastic direction, we discretize the problem using the 4th order Smolyak sparse grid collocation points, getting a total of $M = 3361$ sampling points θ^i . In the offline stage we choose $l = 32$ in (4.4), then \tilde{D} contains all the Fourier modes that can resolved by the given mesh. In Algorithm 4.2, we choose

$$K = 50, \quad \epsilon/(10\sqrt{2/\pi}) = 3 \times 10^{-4}, \quad r = 5.$$

The average number of local stochastic basis functions constructed in the offline stage is

$$k = \frac{1}{n} \sum_{i=1}^n k_i \approx 16.5,$$

which is small given that the stochastic input has high dimension $m = 12$. The maximum number of local stochastic basis functions, $\max_i k_i = 41$, is significantly larger than k . This reveals that the solution space has significantly richer stochastic structure in some regions of the domain than others, and this heterogeneous stochastic structure of the solution space is recognized by the HSFEM.

The forcing function we choose in the online stage is

$$f(x, y) = 1 + x - 2y.$$

The expectation and standard deviation of the solution, and the numerical errors in these two quantities are plotted in Figure 9. From this figure we can see that the HSFEM attains high accuracy in both expectation and standard deviation of the solution. The relative $L^2(D \times \Omega)$ error in the stochastic part of

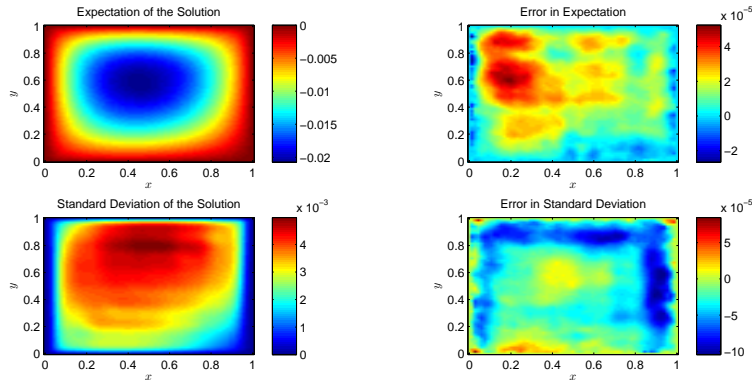


Figure 9: Error in Expectation and Standard Deviation of the solution.

the solution E_{HSFEM} , and relative KL truncation error E_{KL} are listed in Table 2. $E_{HSFEM} = 4.5 \times 10^{-2}$ is small, which means our method attains good accuracy in capturing the stochastic part of the solution, $u(x, \omega) - \bar{u}(x)$. For this problem, E_{HSFEM} is significantly less than E_{KL} , and as we have argued, this reveals the heterogeneous stochastic structure of the solution space and the advantage of approximating the stochastic behavior of the solution using local stochastic basis.

k	E_{HSFEM}	E_{KL}
16.5	4.46×10^{-2}	9.60×10^{-2}

Table 2: The relative $L^2(D \times \Omega)$ error and the relative KL truncation error E_{KL} .

6.4 A 2D Example with Discontinuous Coefficients

In this section, we consider solving equation (6.13) with $a(x, y, \omega)$ discontinuous. We divide $D = [0, 1] \times [0, 1]$ to 4 regions, which are

$$D_1 = [0, 1/2] \times [0, 1/2], \quad D_2 = [1/2, 1] \times [0, 1/2], \quad D_3 = [0, 1/2] \times [1/2, 1], \quad D_4 = [1/2, 1] \times [1/2, 1].$$

And the coefficient $a(x, y, \omega)$ is given by

$$\log(a(x, y, \omega)) = \begin{cases} \sum_{k=1}^3 \omega_k (\sin(2k\pi x) + \cos(2(4-k)\pi y)), & (x, y) \in D_1, \\ 1 + \sum_{k=4}^6 \omega_k (\sin(2(k-3)\pi x) + \cos(2(7-k)\pi y)), & (x, y) \in D_2, \\ 2 + \sum_{k=7}^9 \omega_k (\sin(2(k-6)\pi x) + \cos(2(10-k)\pi y)), & (x, y) \in D_3, \\ 3 + \sum_{k=10}^{12} \omega_k (\sin(2(k-9)\pi x) + \cos(2(13-k)\pi y)), & (x, y) \in D_4, \end{cases} \quad (6.14)$$

where $\omega_i, i = 1, \dots, 12$ are independent standard Gaussian random variables, $\omega_i \sim \mathcal{N}(0, 1)$.

None of the random variables ω_i in (6.14) is negligible, so this problem has real high stochastic dimension. The discretization of the problem in the spatial and stochastic directions are the same as the previous example. And in the randomized range finding Algorithm 4.2 we choose

$$K = 50, \quad \epsilon/(10\sqrt{2/\pi}) = 10^{-4}, \quad r = 5.$$

The average number of local stochastic basis functions constructed in the offline stage is

$$k = \frac{1}{n} \sum_{i=1}^n k_i \approx 12.3,$$

which can be considered as small given that the stochastic input has high dimension $m = 12$.

The maximum number of local stochastic basis functions $\max_i k_i = 40$ is significantly larger than k , which reflects the heterogeneous stochastic structure of the solution space. The distribution of k_i is plotted in Figure 10. We can see that more local stochastic basis functions are put in D_1 , which according to (6.14), is the region where the random part of the coefficient $a(x, \omega)$ has stronger effect.

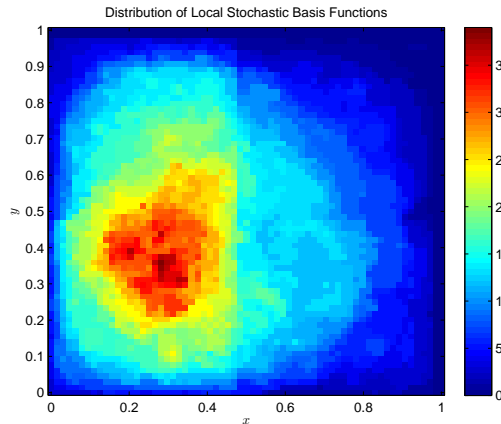


Figure 10: Distribution of local stochastic basis for the example with discontinuous coefficient.

The forcing function we choose in the online stage is $f(x, y) = 1 + x - 2y$. The expectation and standard deviation of the solution, and the numerical errors in these two quantities are plotted in Figure 11. The relative error in stochastic part of the solution E_{HSFEM} , and relative KL truncation error E_{KL} are listed in Table 3. We can see that our method attains good accuracy for this problem with discontinuous coefficients. Again, we find that E_{HSFEM} is significantly smaller than E_{KL} .

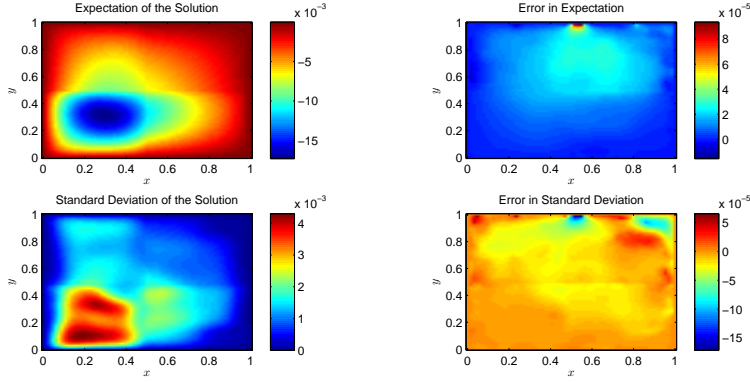


Figure 11: Error in Expectation and Standard Deviation of the solution.

k	E_{HSFEM}	E_{KL}
15.2	3.3×10^{-2}	1.5×10^{-1}

Table 3: The relative $L^2(D \times \Omega)$ error E_{HSFEM} and the relative KL truncation error E_{KL} .

6.5 A 2D Problem With Very High Dimensional Stochastic Input

In this subsection we study a 2D problem with very high dimensional Gaussian input. We denote the following set of orthonormal $L^2([0, 1]^2)$ functions

$$\{2 \sin(2\pi x), \dots, 2 \sin(2k\pi x), \dots, 2 \sin(12\pi x)\} \otimes \{2 \cos(2\pi y), \dots, 2 \cos(2k\pi y), \dots, 2 \cos(12\pi y)\},$$

as $\{\Psi_1(x, y), \Psi_2(x, y), \dots, \Psi_{36}(x, y)\}$, and consider operator $-\operatorname{div}(a(x, y, \omega)\nabla(\cdot))$ with $a(x, y, \omega)$ given by

$$\log(a(x, y, \omega)) = \frac{1}{2} \sum_{i=1}^{36} \omega_i \Psi_i(x, y),$$

where $\omega_i, i = 1, \dots, 36$ are independent standard Gaussian random variables, $\omega_i \sim \mathcal{N}(0, 1)$. This operator has genuine high dimensional stochastic input, and $\omega_i, i = 1, \dots, 36$, contribute equally to $\log(a_k(x, y, \omega))$ in $L^2(D \times \Omega)$. We discretize the domain using a standard right triangular mesh of size $h = \frac{1}{32}$, and D is divided into 2×32^2 elements. We choose $l = 16$ in the discretization of $L^2(D)$ (4.4), then \tilde{D} contains all the Fourier modes that can be resolved by this mesh. In the stochastic direction, since the dimension of ω is 36, which is very high, a very large number of collocation points are required even if we use the Smolyak sparse grid quadrature. So we instead discretize Ω using $M = 10^4$ Monte Carlo sampling points and set the weights $w^i = 1/M$. In Algorithm 4.2, we choose $K = 70$, $\epsilon/(10\sqrt{2/\pi}) = 2 \times 10^{-3}$, $r = 5$. The average number of the local stochastic basis functions constructed offline is

$$k = 17.6.$$

This k is quite small given that this problem has very high stochastic dimension $m = 36$. In addition, $\max_i k_i = 51$ is significantly larger than k , which implies the solution space has strong heterogeneous stochastic structure, and this is recognized by the HSFEM.

In the online stage, we solve the equation using forcing $f(x) = 1 + x - 2y$. The expectation and standard deviation of the solution and numerical errors in these two quantifies are plotted in Figure 12. The relative numerical error E_{HSFEM} and relative KL truncation error are listed in Table 4. $E_{HSFEM} = 1.04 \times 10^{-1}$, which means the HSFEM numerical solution captures about 90% of the stochastic part of the solution. The numerical error E_{HSFEM} is significantly smaller than the KL truncation error E_{KL} .

Recall that in computing the numerical errors, we did not consider the discretization error in (6.1), and we were comparing our numerical solutions with the MC or SC solutions. In this example, the

k	E_{HSFEM}	E_{KL}
17.6	1.04×10^{-1}	4.42×10^{-1}

Table 4: The relative $L^2(D \times \Omega)$ error E_{HSFEM} and the relative KL truncation error E_{KL} .

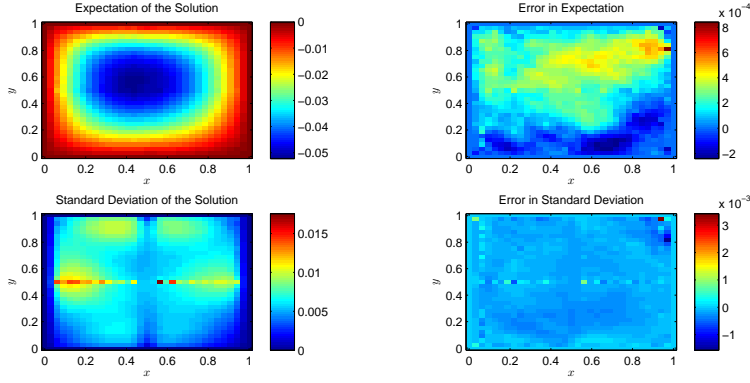


Figure 12: Expectation and Standard Deviation of the solution.

number of MC sampling points $M = 10^4$ is actually not very large, and the discretization error in (6.1) may be dominant. Nevertheless, our numerical results still demonstrate the capacity of the HSFEM in exploiting the local stochastic structure of the solution space to the discretized problem.

7 Concluding Remarks

A new concept of sparsity has been introduced for linear stochastic elliptic operator, which reflects the compactness of its inverse operator in the stochastic direction and allows for spatially heterogeneous stochastic structure of the corresponding solution space. This new concept of sparsity motivates a HSFEM framework for solving linear stochastic elliptic equations, which uses a problem-dependent and local stochastic basis to approximate the solution. This HSFEM framework provides a novel direction to attack the *curse of dimensionality* by exploiting the local stochastic structure of the solutions.

Constructing a suitable local stochastic basis is a challenging task since the inverse of the elliptic operator is highly implicit, nonlinear and non-local. In this work we provide a sampling method to construct the local stochastic basis using the randomized range finding methods. It involves sampling the stochastic operator for K times using randomly chosen forcing functions and some orthogonalization process. Hopefully, if the stochastic operator enjoys the Operator-Sparsity, a small K is sufficient to identify the local stochastic structure of the solution space. The relatively expensive offline computation limits the application of the HSFEM to the multi-query setting. Methods to reduce the offline computational cost and an online error estimation and correction procedure are given.

Numerical results are presented to demonstrate the spatially heterogeneous stochastic structure of the solution space to stochastic elliptic equations, and the Operator-Sparsity for several elliptic operators with high dimensional stochastic input. The proposed HSFEM can recognize and respect the heterogeneous stochastic structure of the solution space, thus achieve high efficiency in the online stage.

We only consider Dirichlet problem in this paper, but our method can be easily generalized to deal with Neumann boundary condition problem. The key idea of the present work is the use of the heterogeneous coupling of spatial basis with local stochastic basis to approximate the solution, which should not be restricted to elliptic equations, and will be applied to time-dependent problems in our future work. Convergence analysis of the HSFEM framework will be given in another paper.

8 Acknowledgements

We would like to thank the two anonymous reviewers for their constructive comments which help improve the quality of this paper. This research was in part supported by Air Force MURI Grant FA9550-09-1-0613, DOE grant DE-FG02-06ER257, and NSF Grants No. DMS-1318377, DMS-1159138.

References

- [1] Ivo Babuška and Panagiotis Chatzipantelidis. On solving elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 191(37):4093–4122, 2002.
- [2] Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM review*, 52(2):317–355, 2010.
- [3] Ivo Babuska, Raúl Tempone, and Georgios E Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.
- [4] Marcel Bieri. A sparse composite collocation finite element method for elliptic spdes. *SIAM Journal on Numerical Analysis*, 49(6):2277–2301, 2011.
- [5] Marcel Bieri, Roman Andreev, and Christoph Schwab. Sparse tensor discretization of elliptic spdes. *SIAM Journal on Scientific Computing*, 31(6):4281–4304, 2009.
- [6] Marcel Bieri and Christoph Schwab. Sparse high order fem for elliptic spdes. *Computer Methods in Applied Mechanics and Engineering*, 198(13):1149–1170, 2009.
- [7] Russel E Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.
- [8] Robert H Cameron and William T Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.
- [9] Mulin Cheng, Thomas Y Hou, Mike Yan, and Zhiwen Zhang. A data-driven stochastic method for elliptic pdes with random coefficients. *Journal on Uncertainty Quantification*, 1, 2013.
- [10] Mulin Cheng, Thomas Y Hou, and Zhiwen Zhang. A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations i: Derivation and algorithms. *Journal of Computational Physics*, 242:843–868, 2013.
- [11] Mulin Cheng, Thomas Y Hou, and Zhiwen Zhang. A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations ii: Adaptivity and generalizations. *Journal of Computational Physics*, 242:753–776, 2013.
- [12] Alireza Doostan, Roger G Ghanem, and John Red-Horse. Stochastic model reduction for chaos representations. *Computer Methods in Applied Mechanics and Engineering*, 196(37):3951–3966, 2007.
- [13] Alireza Doostan and Houman Owhadi. A non-adapted sparse approximation of pdes with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.
- [14] Philipp Frauenfelder, Christoph Schwab, and Radu Alexandru Todor. Finite elements for elliptic problems with stochastic coefficients. *Computer methods in applied mechanics and engineering*, 194(2):205–228, 2005.
- [15] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*, volume 41. Springer, 1991.
- [16] Michael Grüter and Kjell-Ove Widman. The green function for uniformly elliptic equations. *Manuscripta Mathematica*, 37(3):303–342, 1982.
- [17] Wolfgang Hackbusch. A sparse matrix arithmetic based on h-matrices. part i: Introduction to h-matrices. *Computing*, 62(2):89–108, 1999.
- [18] Wolfgang Hackbusch and Boris N Khoromskij. A sparse h-matrix arithmetic. part ii: application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.
- [19] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- [20] Thomas Y Hou, Wuan Luo, Boris Rozovskii, and Hao-Min Zhou. Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics. *Journal of Computational Physics*, 216(2):687–706, 2006.

- [21] Kazufumi Ito and SS Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal of computational physics*, 143(2):403–425, 1998.
- [22] Michael Kleiber and Tran Duong Hien. *The stochastic finite element method: basic perturbation technique and computer implementation*. Wiley New York, 1992.
- [23] Mathelin L. and Gallivan K.A. A compressed sensing approach for partial differential equations with random input data. *Communications in computational physics*, 12(4):919–954, 2012.
- [24] Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007.
- [25] Lin Lin, Jianfeng Lu, and Lexing Ying. Fast construction of hierarchical matrix representation from matrix–vector multiplication. *Journal of Computational Physics*, 230(10):4071–4087, 2011.
- [26] Lionel Mathelin, M Yousuff Hussaini, and Thomas A Zang. Stochastic approaches to uncertainty quantification in cfd simulations. *Numerical Algorithms*, 38(1-3):209–236, 2005.
- [27] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. SIAM, 1992.
- [28] Fabio Nobile, Raúl Tempone, and Clayton G Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.
- [29] Gianluigi Rozza, DBP Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.
- [30] Khachik Sargsyan, Cosmin Safta, Habib N Najm, Bert J Deusschere, Daniel Ricciuto, and Peter Thornton. Dimensionality reduction for complex models via bayesian compressive sensing. *International Journal for Uncertainty Quantification*, 4(1), 2014.
- [31] Sergey A Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Dokl. Akad. Nauk SSSR*, volume 4, page 123, 1963.
- [32] Henry Stark and John William Woods. *Probability, random processes, and estimation theory for engineers*. Prentice-Hall Englewood Cliffs (NJ), 1986.
- [33] Olaf Steinbach. *Numerical approximation methods for elliptic boundary value problems: Finite and boundary elements*, volume 99. Springer, 2008.
- [34] JL Taylor, S Kim, and RM Brown. The green function for elliptic systems in two dimensions. *Communications in Partial Differential Equations*, 38(9):1574–1600, 2013.
- [35] Norbert Wiener. The homogeneous chaos. *Amer. J. Math*, 60(4):897–936, 1938.
- [36] Dongbin Xiu and Jan S Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.
- [37] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.